Configuring Frame-Mode MPLS on IOS Platforms

Overview

This module commences with a brief description of Cisco's switching implementations and describes the Cisco IOS implementation of frame-mode MPLS. It gives detailed configuration, monitoring and debugging guidelines.

It includes the following topics:

- CEF Switching Review
- Configuring MPLS
- Configuring Frame-Mode MPLS on Switched WAN Media
- Monitoring MPLS on Cisco IOS—Frame-Mode Interfaces

Objectives

Upon completion of this module, the learner will be able to perform the following tasks:

- Explain the basics of CEF switching
- Understand how to configure MPLS on frame-mode MPLS interfaces on IOS platforms
- Explain advanced core MPLS configuration options (TTL propagation, controlled label distribution) on IOS platforms
- Understand how to configure frame-mode MPLS on ATM PVC on IOS platforms
- Monitor and troubleshoot MPLS on IOS platforms

CEF Switching Review

Objectives

Upon completion of this lesson, the learner will be able to perform the following tasks:

- Describe the CEF switching mechanism
- Identify the difference between CEF switching and other switching mechanisms
- Explain how to configure CEF switching on IOS platforms
- Monitor and troubleshoot CEF switching

IOS Switching Mechanisms

IOS supports three IP switching mechanisms:

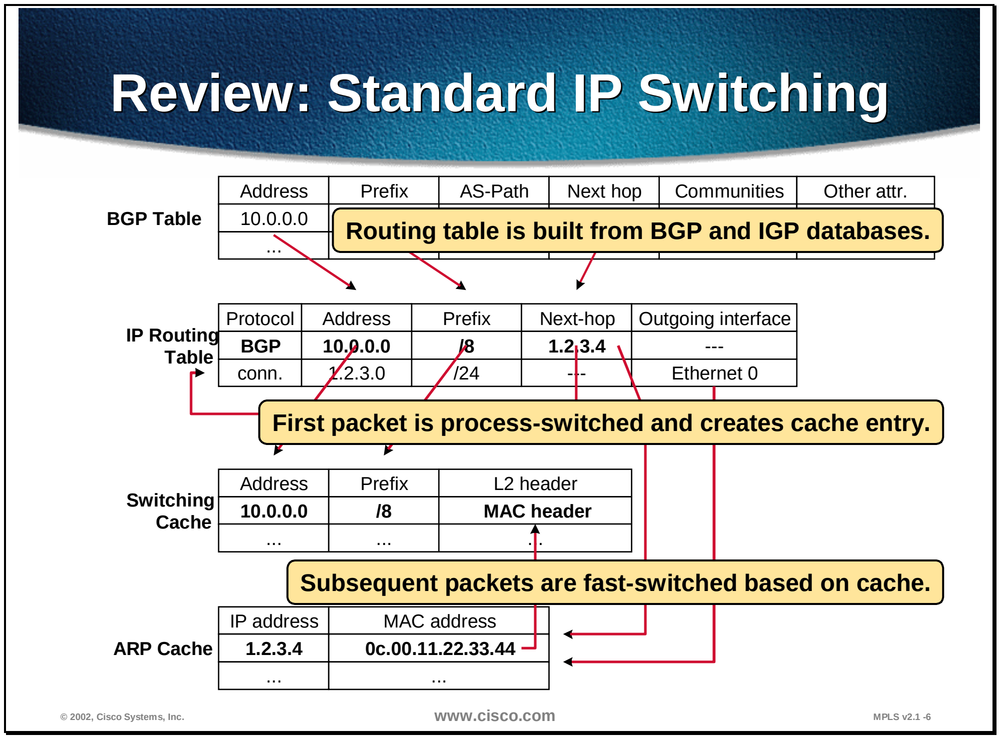
- **Routing-table driven switching (full lookup at every packet)—process switching**
- **Cache-driven switching (most recent destinations are entered in the cache):**
  - **Fast switching, Optimum switching, etc.**
  - **First packet is always process switched**
- **Topology-driven switching:**
  - **Cisco Express Forwarding (CEF):**
    - **Forwarding table (FIB) is pre-built**

 www.cisco.com MPLS v2.1 -5

The first and the oldest switching mechanism available in Cisco routers is called Process Switching. Because it has to find a destination in the routing table (possibly a recursive lookup) and construct a new Layer 2 frame header for every packet it is very slow and is normally not used.

To overcome the slow performance of process switching, Cisco IOS® supports several switching mechanisms that use a cache to store the most recently used destinations. The cache uses a faster searching mechanism and it also stores the entire Layer 2 frame header to improve the encapsulation performance. The first packet whose destination is not found in the fast-switching cache is process-switched and an entry is created in the cache. The following packets are switched in the interrupt code using the cache to improve performance.

The latest and the preferred Cisco IOS switching mechanism is the Cisco Express Forwarding (CEF), which incorporates the best of all the previous switching mechanisms. Packets are switched in the interrupt code using the CEF cache (FIB table). It supports per-packet load balancing (previously only supported by process switching), per-source/destination load balancing (only supported by CEF switching), fast destination look-up and many other features not supported by other switching mechanisms. The FIB table is essentially a replacement for the standard routing table.



The figure shows a sequence of events when process switching and fast switching is used for destinations learned through the Border Gateway Protocol (BGP).

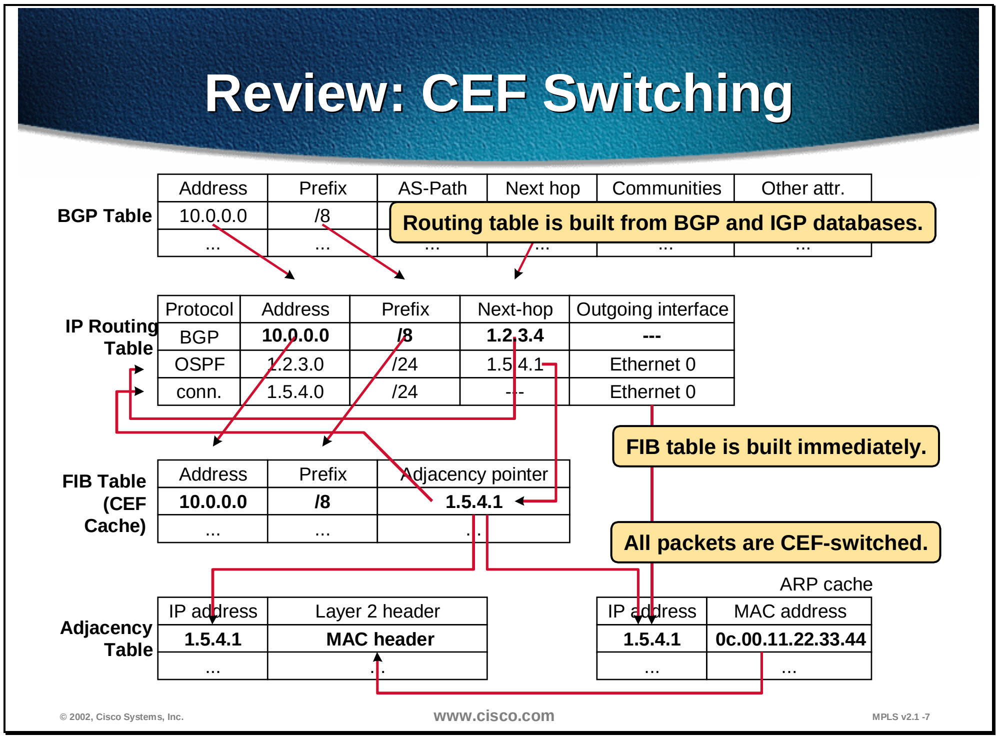
When a BGP update is received and processed an entry is created in the routing table.

When the first packet arrives for this destination the router tries to find the destination in the fast-switching cache. Since it is not there, the process switching has to switch the packet when the process is run. The process performs a recursive look-up to find the outgoing interface. It may possibly trigger an ARP request or find the Layer 2 address in the ARP cache. Finally it creates an entry in the fast-switching cache.

All subsequent packets for the same destination are fast-switched:

- The switching occurs in the interrupt code (the packet is processed immediately)
- Fast destination look-up is performed (no recursion)
- The encapsulation uses a pre-generated Layer 2 header that contains the destination as well as the source Layer 2 (MAC) address (no ARP request or ARP cache look-up is necessary)

Note Whenever a router receives a packet that should be fast-switched, but the destination is not in the switching cache, the packet is process-switched—full routing table lookup is performed and an entry in the fast-switching cache is created to ensure that the subsequent packets for the same destination prefix will be fast-switched.

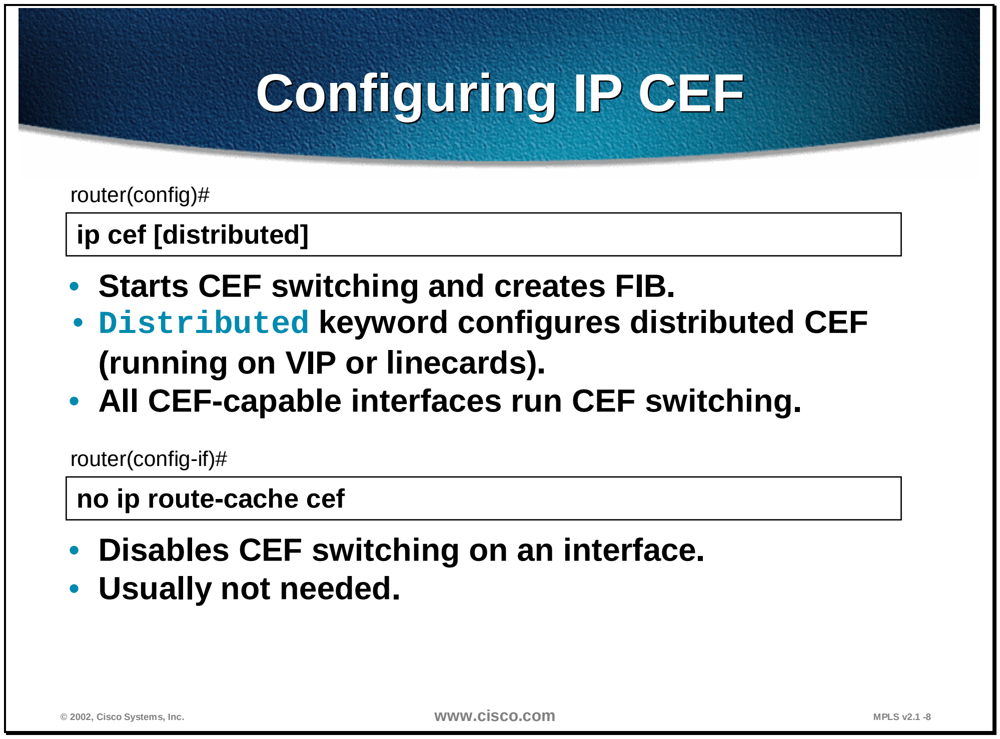


The CEF uses a different architecture than either process-switching or any other cache-based switching mechanism. The CEF uses a complete IP switching table, called a Forwarding Information Base (FIB), which holds the same information as the routing table. The generation of entries in the FIB table is not traffic driven (packet triggered) but topology driven (change triggered)—when something changes in the routing table, the change is also reflected in the FIB table.

Note As the FIB contains complete IP switching table, the router can make a definitive decisions based on FIB. Whenever a router receives a packet that should be CEF switched, but the destination is not in the FIB, the packet is dropped.

The FIB table is also different from other fast-switching caches in that it does not contain the information about the outgoing interface and the corresponding Layer 2 header. That information is stored in a separate table—the Adjacency table. This table is more or less a copy of the ARP cache but instead of holding only the destination MAC address it holds the Layer 2 header.

Note If the router carries full Internet routing (around 120.000 networks at the time of writing), enabling the CEF may consume additional memory. Enabling the distributed CEF will also impact memory utilisation on VIP modules (Cisco 7500) or linecards (Cisco 12000), since the entire FIB table is copied to all VIP modules or linecards.



ip cef

To enable the CEF on the route processor card, use the **ip cef** global configuration command. To disable the CEF, use the **no** form of this command.

ip cef [**distributed**]
no ip cef [**distributed**]

Syntax Description

distributed	(Optional) Enables the distributed CEF (dCEF) operation. Distributes the CEF information to the line cards. The line cards perform express forwarding.

Defaults

On this platform...	The default is...
Cisco 7000 series equipped with RSP7000	CEF is not enabled
Cisco 7200 series	CEF is not enabled
Cisco 7500 series	CEF is not enabled
Cisco 12000 series Gigabit Switch Router	Distributed CEF is enabled

ip route-cache cef

To enable the CEF operation on an interface after the CEF operation has been disabled, use the **ip route-cache cef** interface configuration command. To disable the CEF operation on an interface, use the **no** form of this command.

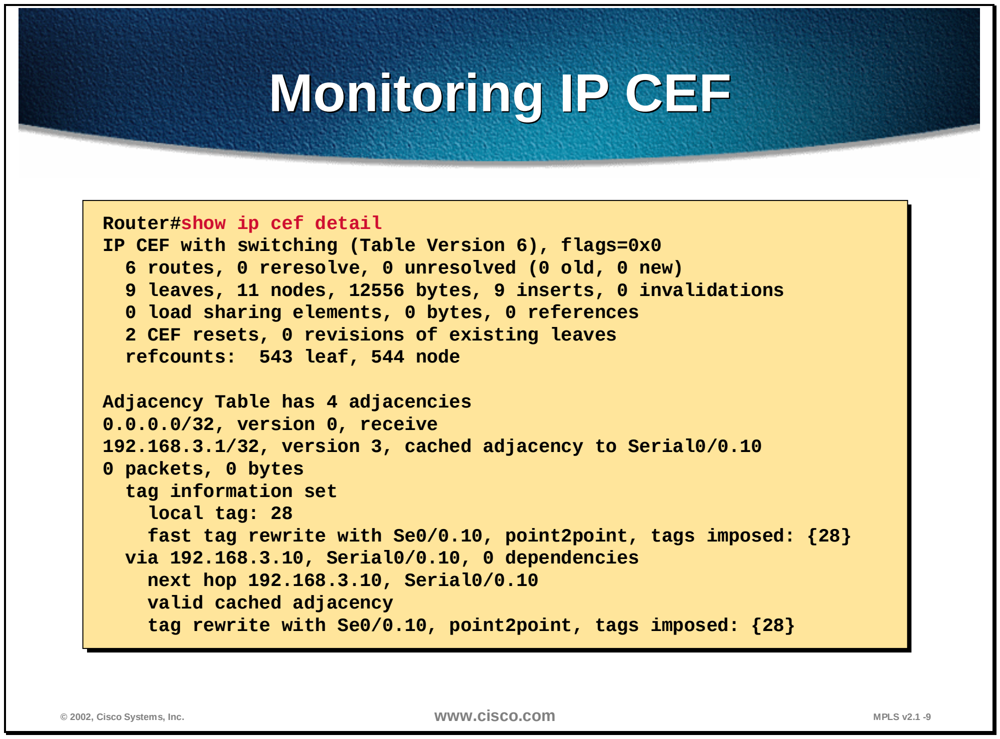
ip route-cache cef
no ip route-cache cef

Syntax Description

This command has no arguments or keywords.

Defaults

When standard CEF or dCEF operations are enabled globally, all interfaces that support the CEF are enabled by default.



show ip cef

To display entries in the FIB that are unresolved, or to display a summary of the FIB, use the following form of the **show ip cef** EXEC command:

show ip cef [**unresolved** | **summary**]

To display specific entries in the FIB based on IP address information, use the following form of the **show ip cef** EXEC command:

show ip cef [***network*** [***mask*** [**longer-prefix**]]] [**detail**]

To display specific entries in the FIB based on interface information, use the following form of the **show ip cef** EXEC command:

show ip cef [***type number***] [**detail**]

Syntax Description

unresolved	(Optional) Displays unresolved FIB entries.
summary	(Optional) Displays a summary of the FIB.
network	(Optional) Displays the FIB entry for the specified destination network.
mask	(Optional) Displays the FIB entry for the specified destination network and mask.
longer-prefix	(Optional) Displays FIB entries for all the more specific destinations.
detail	(Optional) Displays detailed FIB entry information.

type number	(Optional) Interface type and number for which to display FIB entries.

Summary

CEF switching is one of many different switching mechanisms supported by Cisco IOS software. It combines good performance with support for the advanced features needed in modern networks.

The CEF switching table (FIB table) contains all the information that is in the routing table and, therefore, is not a packet-triggered cache mechanism. On the other hand it incorporates a fast lookup mechanism to provide excellent performance.

Use the **ip cef** global command to enable CEF switching on all interfaces that support it. Use the **no ip route-cache cef** interface command to disable CEF switching on an interface.

Use the **show ip cef** command to view the contents of the FIB table and **show adjacency** to view the contents of the adjacency table.

Lesson Review

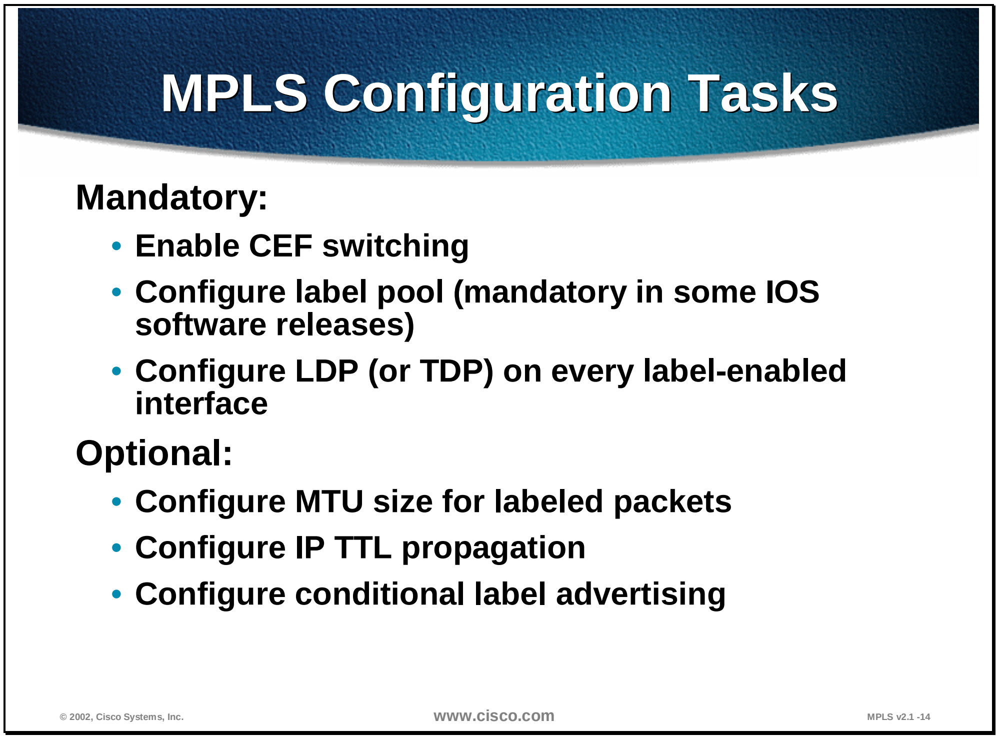
1. What are the three types of Layer 3 switching mechanisms available in IOS?
2. Identify at least two cache-based switching mechanisms in IOS.
3. What is the main difference between fast switching and Cisco Express Forwarding?
4. What are the main data structures used by CEF?
5. What happens if the destination subnet is not in the fast-switching cache?
6. What happens if the destination subnet is not in the Forwarding Information Base?

Configuring MPLS

Objectives

Upon completion of this lesson, the learner will be able to perform the following tasks:

- Describe the configuration required to enable MPLS on frame-mode interfaces
- Understand the IP TTL Propagation configuration option
- Explain LAN MTU size issues
- Understand appropriate MPLS configuration parameters
- Explain how to configure controlled label distribution

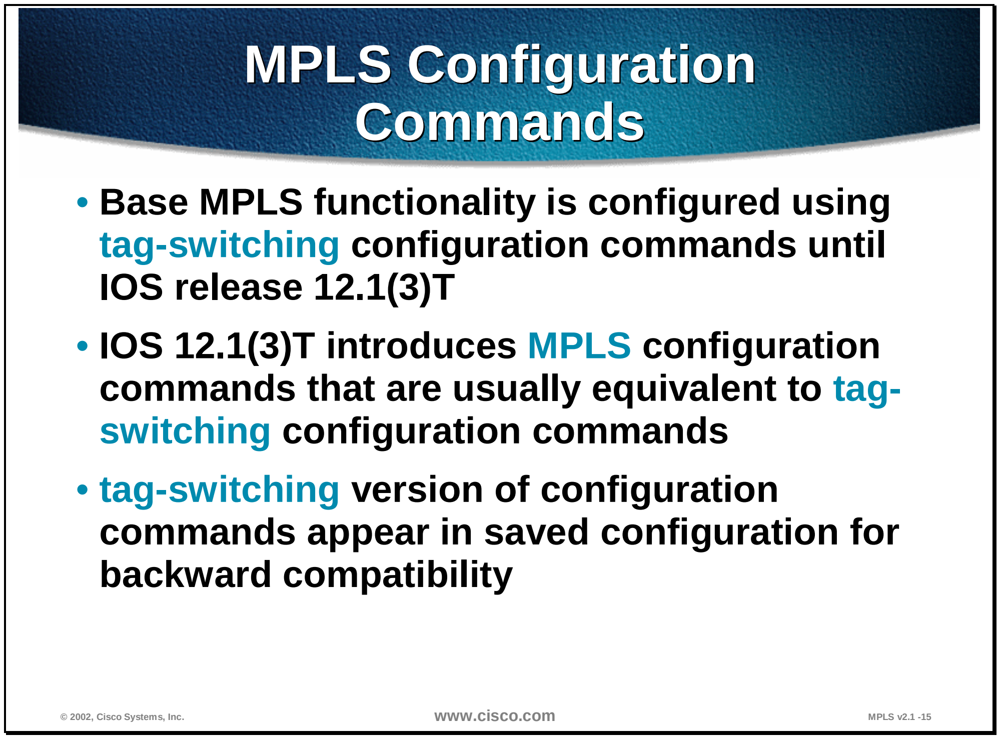


To enable MPLS the CEF switching must first be enabled. Depending on the IOS version a pool of labels may need to be set. To enable LDP (or TDP) enable either the label switching or the tag switching on the interface. Optionally, the maximum size (MTU) of labeled packets may be changed.

By default, the TTL field is copied from the IP header into the label when a packet is entering a MPLS network. To prevent core routers from responding with "ICMP time to live exceeded" messages, disable the TTL propagation. If the TTL propagation is disabled, 255 is the value contained in the TTL field of the label.

Note Ensure that all routers have TTL propagation either enabled or disabled. If some routers have TTL enabled and others disabled the result may be that a packet leaving the MPLS domain will have a larger TTL value than when it entered.

By default, a router will generate and propagate labels for all networks that it has in the routing table. If only label switching for a limited number of networks is required (for example, only for router loopback addresses), configure the conditional label advertising.

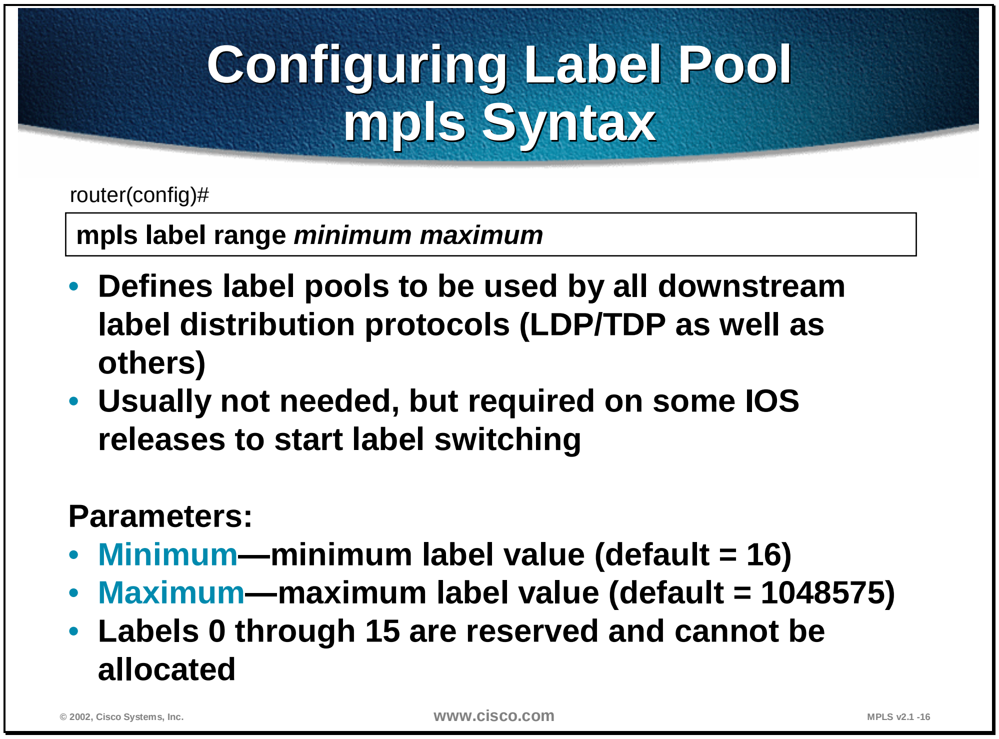


Pre-standard implementation of MPLS on IOS platforms was called **tag switching**. All configuration, debugging and monitoring commands used to configure MPLS on IOS therefore used the **tag-switching** keyword.

Tag switching complies with all IETF standard documents describing MPLS, apart from the label distribution protocol. In this instance, tag switching uses the Tag Distribution Protocol (TDP) whereas the IETF standard documents specify another protocol called Label Distribution Protocol (LDP).

Most of the MPLS configuration, debugging and monitoring commands on IOS platforms were changed (by replacing the **tag-switching** keyword with the **mpls** keyword) to comply with the IETF terminology in IOS release 12.1(3)T, except for the commands that configure TDP parameters. TDP configuration commands still use the old syntax.

The **mpls** commands are usually functionally equivalent to **tag-switching** commands (with a few exceptions noted in this module) and can be used interchangeably. The saved router configurations still contain **tag-switching** commands if at all possible (even if the parameter was configured with **mpls** command) to ensure backward compatibility with older IOS releases.



mpls label range

To configure the range of local labels available for use on packet interfaces, use the **mpls label range** global configuration command. Use the **no** form of this command to revert to the platform defaults.

mpls label range *min max*
no mpls label range

Syntax Description

min	The smallest label allowed in the label space. The default is 16.
max	The largest label allowed in the label space. The default is 1048575.

Defaults

The default values for the arguments of this command are:

- *min*—16
- *max*—1048575

The labels 0 through 15 are reserved by the IETF (see draft-ietf-mpls-label-encaps-07.txt for details) and cannot be included in the range specified by the **mpls label range** command.

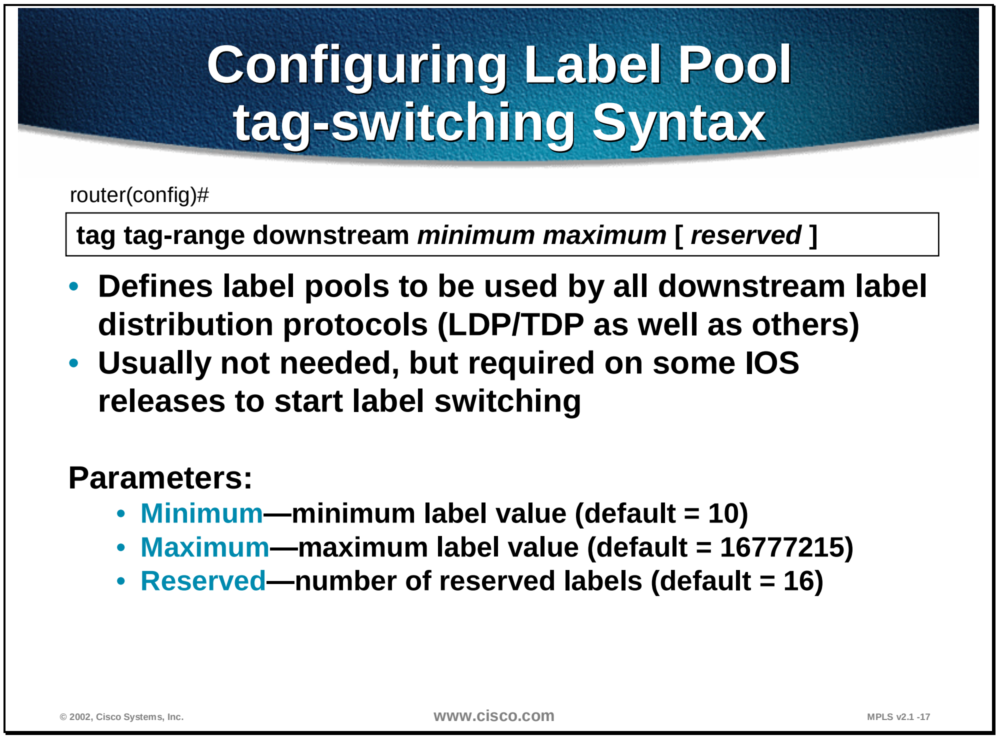
Command Modes

Global configuration

Usage Guidelines

The label range defined by the **mpls label range** command is used by all MPLS applications that allocate local labels (for dynamic label switching, MPLS traffic engineering, MPLS VPNs, etc.)

If a new label range is specified that does not overlap the range currently in use, the new range will not take effect until the router is reloaded again.



tag-switching tag-range downstream

To configure the size of the label (tag) space for downstream unicast label allocation, use the **tag-switching tag-range downstream** command in the global configuration mode. To revert the platform defaults, use the **no** form of this command.

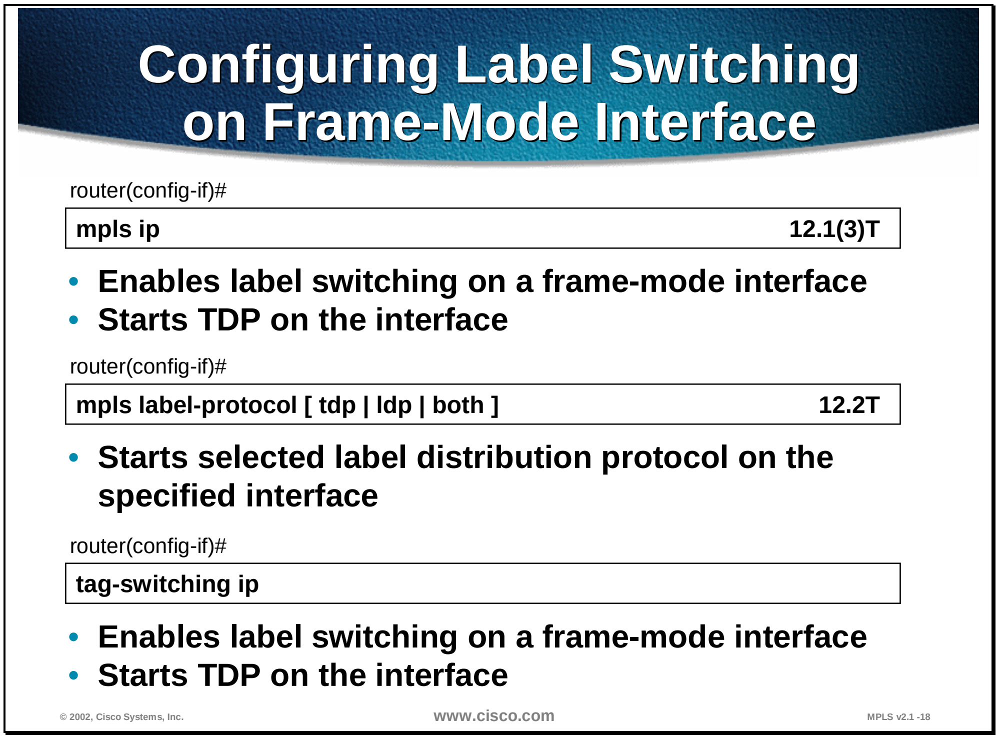
tag-switching tag-range downstream *min max reserved*
no tag-switching tag-range downstream *min max reserved*

Syntax Description

min	The smallest label allowed in the label space. The default is 10.
max	The largest label allowed in the label space. The default is 10000.
reserved	The number of labels reserved for diagnostic purposes. These labels come out of the low end of the label space. Default is 16.

Defaults

min—10
max—10000
reserved—16



mpls ip

To enable MPLS forwarding of IPv4 packets along normally routed paths for a particular interface, use the **mpls ip** interface configuration command. Use the **no** form of the command to disable this feature.

mpls ip
no mpls ip

Syntax Description

This command has no optional keywords or arguments.

Defaults

MPLS forwarding of IPv4 packets along normally routed paths for the interface is disabled.

mpls label-protocol [tdp | ldp | both]

To select the label distribution protocol to be used on an interface, use the **mpls label-protocol** command. Use the **no** form to revert to the default label distribution protocol.

mpls label-protocol ***protocol***
no mpls label-protocol ***protocol***

Syntax Description

This command has one argument:

tdp	Enables TDP on an interface

ldp	Enables LDP on an interface
both	Enables TDP and LDP on an interface

Defaults

The Tag Distribution Protocol is the default protocol.

tag-switching ip

To enable label switching of IPv4 packets on an interface, use the **tag-switching ip** command in interface configuration mode. To disable IP label switching on this interface, use the **no** form of this command.

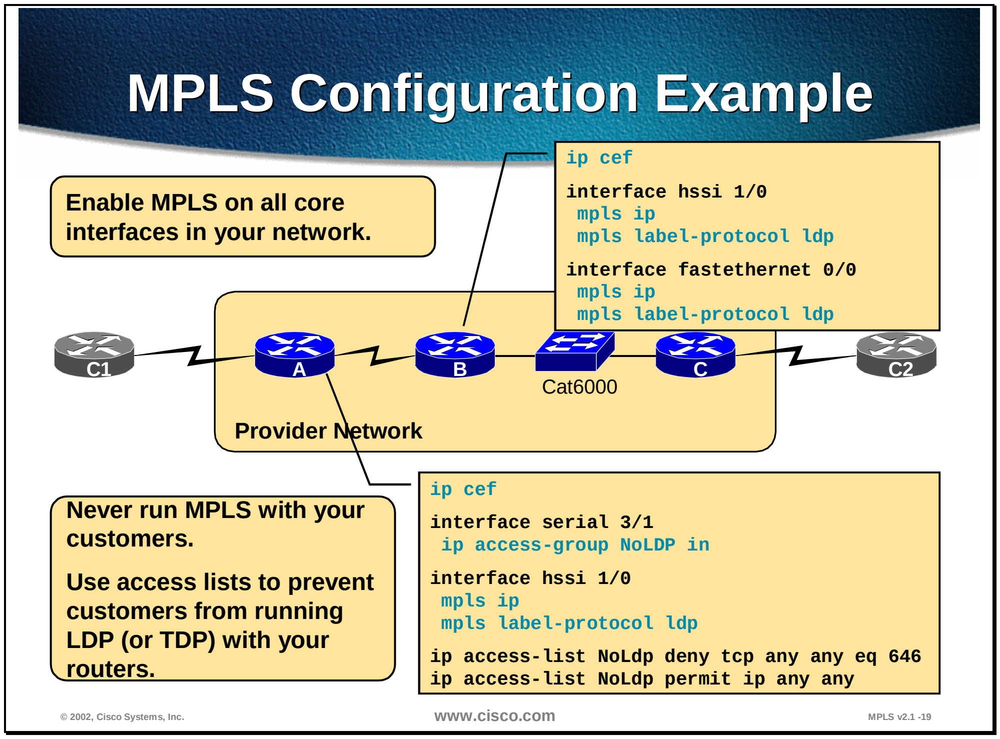
tag-switching ip
no tag-switching ip

Syntax Description

This command has no arguments or keywords.

Defaults

Label switching of IPv4 packets is disabled on this interface.

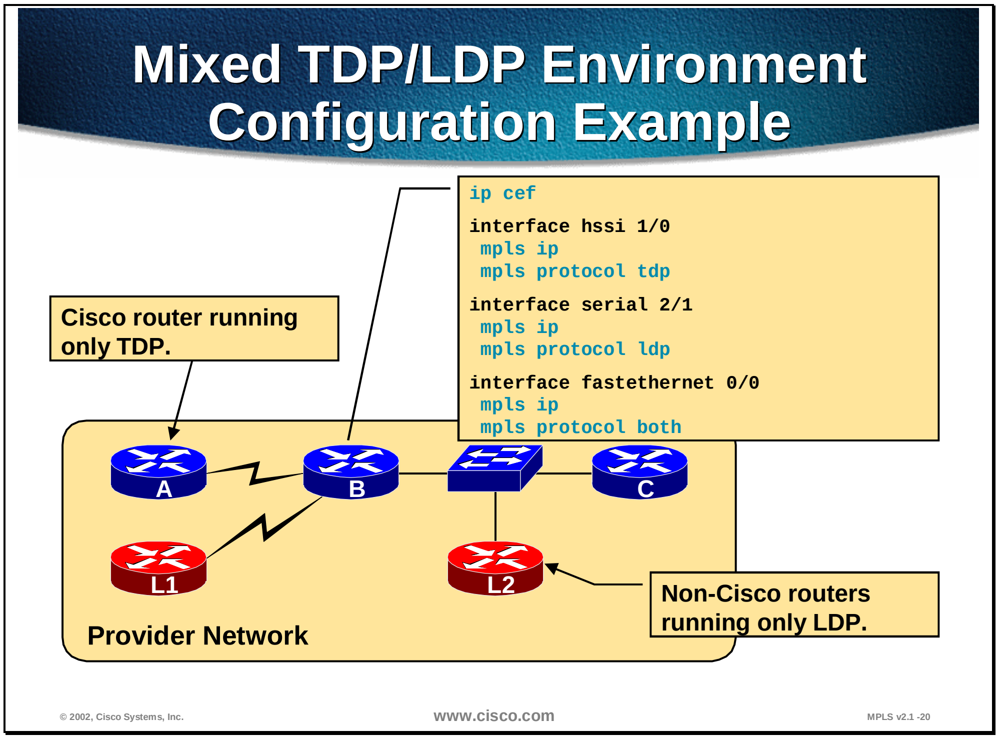


The figure shows the configuration steps needed to enable MPLS on an edge Label Switch Router (LSR). It includes an access list that denies any attempt to establish a LDP session from an interface that is not enabled for MPLS.

It is mandatory to globally enable CEF switching, which automatically enables CEF on all interfaces that support it (CEF is not supported on logical interfaces; for example, loopback interfaces).

All core interfaces have MPLS enabled either by using the **mpls ip** command or, for backwards compatibility, the **tag-switching ip** command. LDP is enabled on all interfaces where mpls is enabled. The **mpls label-protocol ldp** command is required to do that since TDP is the default label protocol.

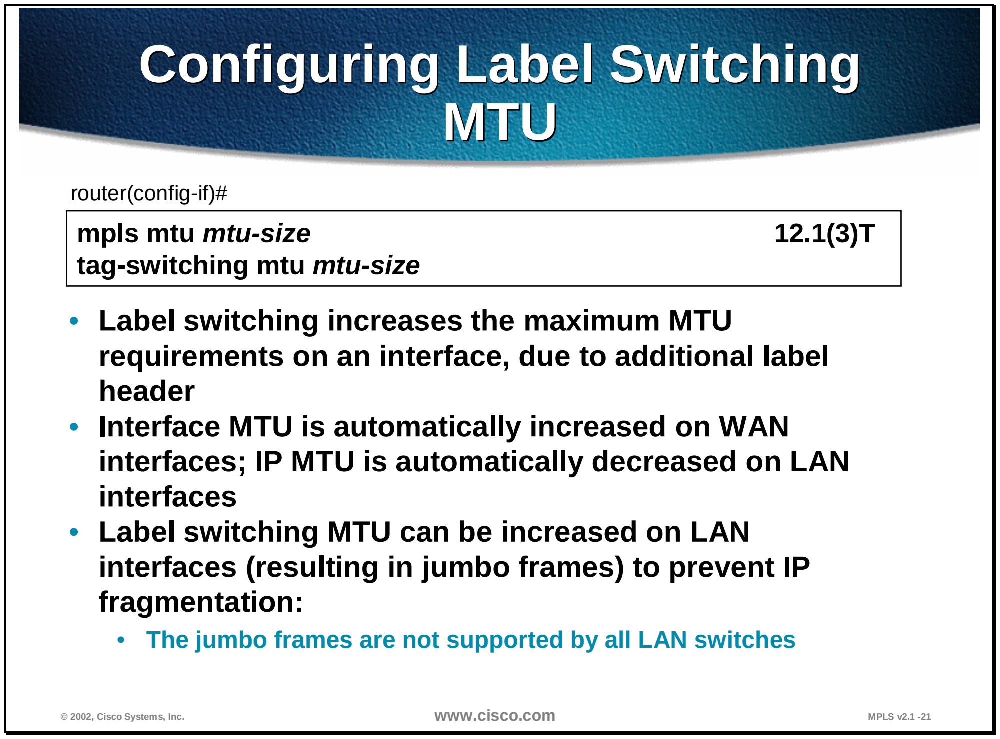
Non-backbone interfaces have an input access list that denies TCP sessions on the well-known port number 646 (LDP). If the TDP protocols would have been used instead of LDP, then the well-known port to protect would have been TCP 711.



The network in the figure consists of routers with different capabilities. Routers A and C are Cisco routers with older software, which supports only TDP as the label protocol. Router L1 and L2 are non-Cisco routers that only support the LDP protocol. Router B, however, is a Cisco router with recent software, which supports both LDP and TDP.

Router B is using TDP on the point-to-point link to router A while it at the same time is using LDP on the other point-to-point link to router L1. Label switching is more or less independent of the distribution protocol so there should be no problem mixing the two protocols. TDP and LDP are functionally very similar.

Both LDP and TDP are used on the shared media (Fast Ethernet). This enables router B to use TDP with router C and LDP with router L2. However, such scenario requires great care since router C and router L2 are not using any common protocol. They will not form any label protocol session. The routing protocol between C and L2 is operating independently of label protocols, so C may very well select L2 as the next-hop for some destinations. But C will not have any labels from L2. This will result in packets being exchanged between C and L2 without labels. Some applications, like MPLS/VPN, will not work in that case.



The commands **mpls mtu** and **tag-switching mtu** are equivalent and can be used interchangeable.

mpls mtu

To override the per-interface maximum transmission unit (MTU), use the **mpls mtu** command in interface configuration mode. To restore the default, use the **no** form of this command.

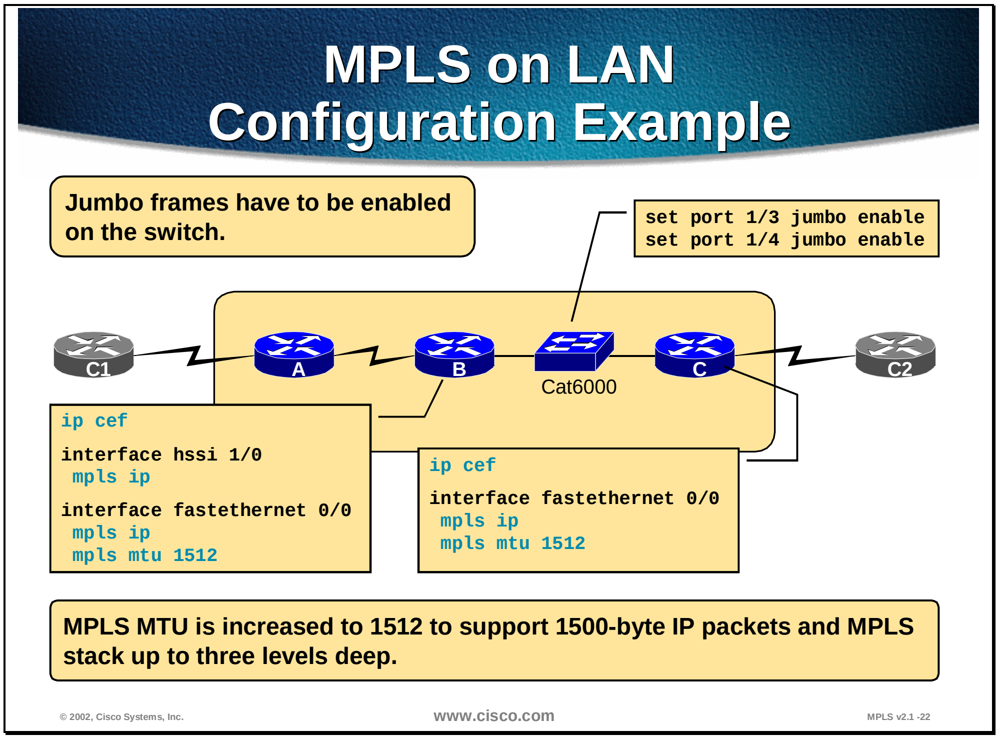
mpls mtu *bytes*
no mpls mtu

Syntax Description

bytes	MTU in bytes.

Defaults

Minimum is 128 bytes. The maximum depends on the type of interface medium.



One way of preventing labeled packets from exceeding the maximum size (and being fragmented as a result) is to increase the MTU of labeled packets for all segments in the label switched path. The problem will typically occur on LAN switches where it is more likely that a device does not support oversized packets (also called *jumbo frames*). Some devices support jumbo frames, some need to be configured to support them.

The MPLS MTU size is increased automatically on WAN interfaces and needs to be increased manually on LAN interfaces. The MPLS MTU size has to be increased on all LSRs attached to a LAN segment. Additionally, the LAN switches used to implement switched LAN segments need to be configured to support jumbo frames. No additional configuration is necessary for shared LAN segments implemented with hubs.

A different approach is needed if a LAN switch does not support jumbo frames (sometimes also called giants or baby giants). The problem may be even worse for networks that do not allow ICMP MTU discovery messages to be forwarded to sources of packets and if they strictly use the DF bit (“do not fragment”). This situation can be encountered where firewalls are used.

Configuring IP TTL Propagation

router(config)#

no mpls ip propagate-ttl **no tag-switching ip propagate-ttl**	**12.1(3)T**

- **By default, IP TTL is copied into label header at label imposition and label TTL is copied into IP TTL at label removal**
- **This command disables IP TTL and label TTL propagation:**
  - **TTL value of 255 is inserted in the label header**
- **The TTL propagation has to be disabled on ingress and egress edge LSR**

 www.cisco.com MPLS v2.1 -23

A label (or label stack) is imposed by the edge LSR when an IP packet is forwarded into the MPLS domain. This label must have a value in the TTL field. By default the edge LSR reads the TTL field in the IP header of the incoming packet, decrements it by one (as always) and copies what is left into the TTL field of the MPLS header.

The core LSRs are only looking at the top most label. They are reading the TTL field of that header and decrements it by one. If the TTL value does not reach zero, the packet is forwarded.

The egress edge LSR that pops the label copies what is left in the label TTL field into the TTL field of the IP header and then forwards the IP packet outside the MPLS domain.

This default behavior can be changed by the **no mpls ip propagate-ttl** configuration command. The ingress edge LSR will use the value 255 as the TTL value in the label when imposing it. The egress edge LSR will not copy the label TTL value to the IP header when popping the label. The net result is that the IP header TTL value will not reflect the hops taken across the MPLS core.

It is important to disable TTL propagation in both ingress and egress edge LSR. Otherwise the IP header may have a higher TTL value when it leaves the MPLS domain than it had when it entered it.

The commands **mpls ip propagate-ttl** and **tag-switching ip propagate-ttl** are equivalent and can be used interchangeable.

mpls ip propagate-ttl

To set the time-to-live (TTL) value on output when the IP packets are being encapsulated in MPLS, use the **mpls ip propagate-ttl** privileged EXEC command. Use the **no** form of the command to disable this feature.

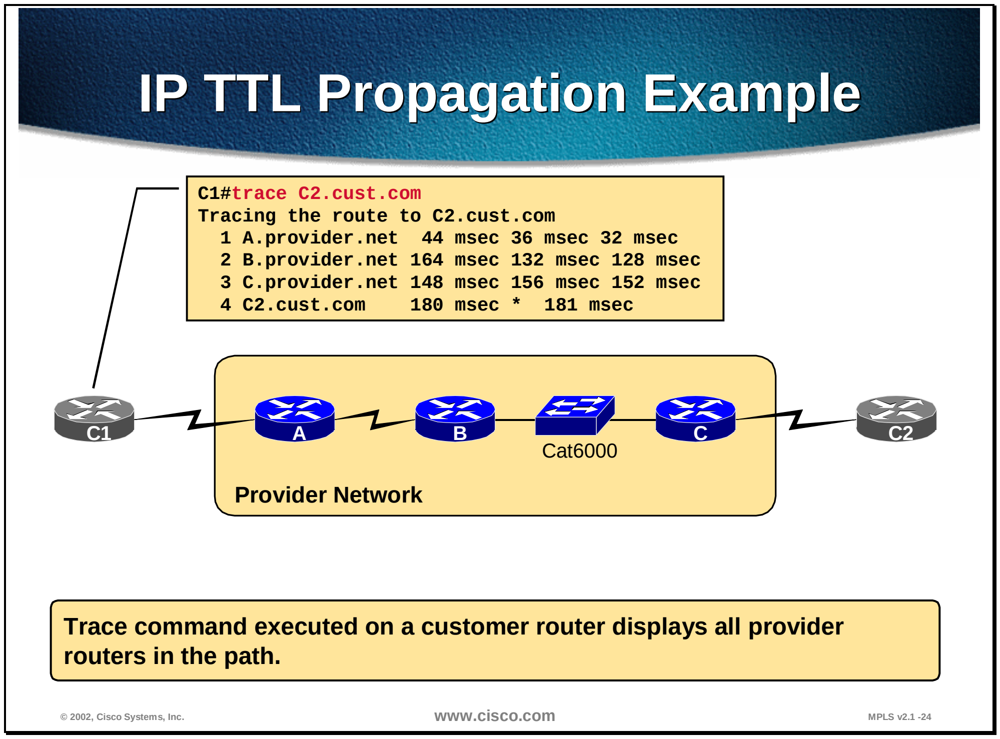
mpls ip propagate-ttl
no mpls ip propagate-ttl

Syntax Description

This command has no optional keywords or arguments.

Defaults

The MPLS TTL value on packet output is set based on the IP TTL value on packet input.



The figure illustrates a typical traceroute behavior in an MPLS network. As the label header of a labeled packet carries the TTL value from the original IP packet, the routers in the path can drop packets for which the time-to-live is exceeded. Traceroute will, therefore, show all the routers in the path. This is the default behavior.

In the example, router C1 is executing a traceroute command that results in the following behavior:

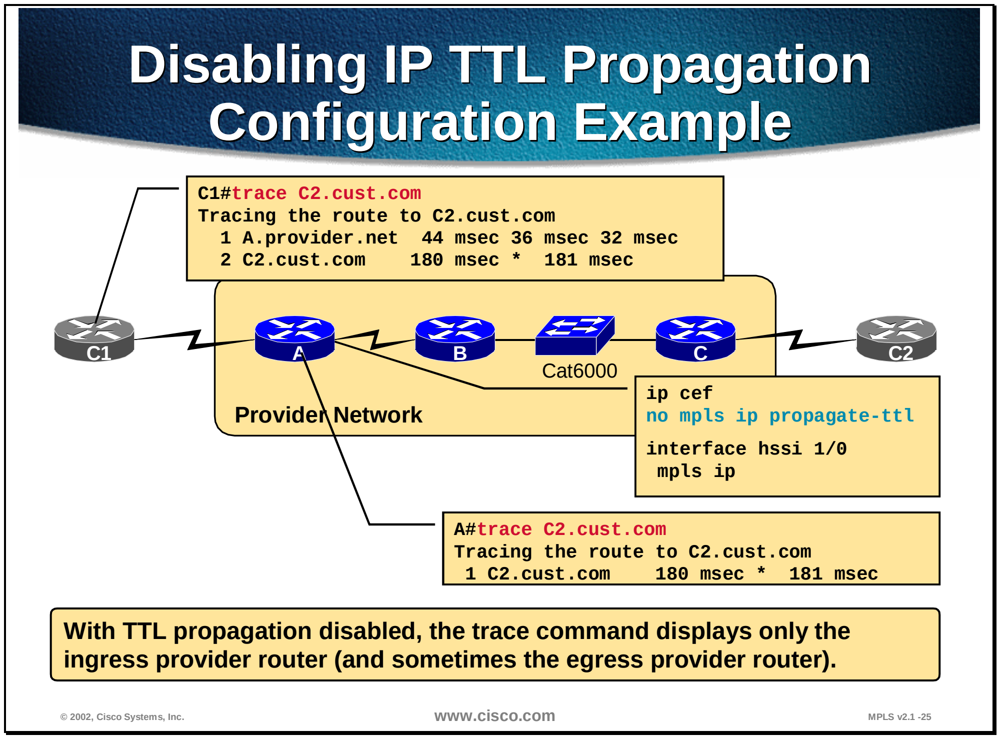
Step 1 The first packet is an IP packet with TTL=1. Router A decreases the TTL and drops the packet because it reaches zero. An ICMP TTL-Exceeded message is sent to the source.

Step 2 The second packet sent is an IP packet with TTL=2. Router A decreases the TTL, labels the packet (the TTL from the IP header is copied into the label) and forwards the packet to router B.

Step 3 Router B decreases the TTL value in the MPLS header, drops the packet and sends an ICMP TTL-Exceeded message to the source. Since it was an MPLS packet that was dropped, the return address for the ICMP message must be derived from the source address in the IP header inside the MPLS packet. But that IP address may actually not be known to router B, so router B forwards the ICMP message along the same label switched path (LSP) as the dropped packet was traveling (in directions towards router C). At the end of the LSP, the label is removed and the ICMP message forwarded according to the destination address in the IP header (towards router C1).

Step 4 The third packet (TTL=3) experiences similar processing as the previous packets, except that router C is now the one dropping the packet based on the TTL in the IP header. Router B, because of penultimate hop popping, previously removed the label and the TTL was copied back to the IP header (or second label).

Step 5 The fourth packet (TTL=4) reaches the final destination where the TTL of the IP packet is examined.



If TTL propagation is disabled, the TTL value is not copied into the label header. Instead, the label's TTL field is set to 255. The result is that no router in the TTL field in the label header will probably be decremented to zero inside the MPLS domain (unless there is a forwarding loop inside the MPLS network).

If the traceroute command is used, ICMP replies are only received from those routers that see the real TTL stored in the IP header.

In the example, router C1 is, again, executing the traceroute command, but the core routers do not copy the TTL to and from the label. This results in the following behavior:

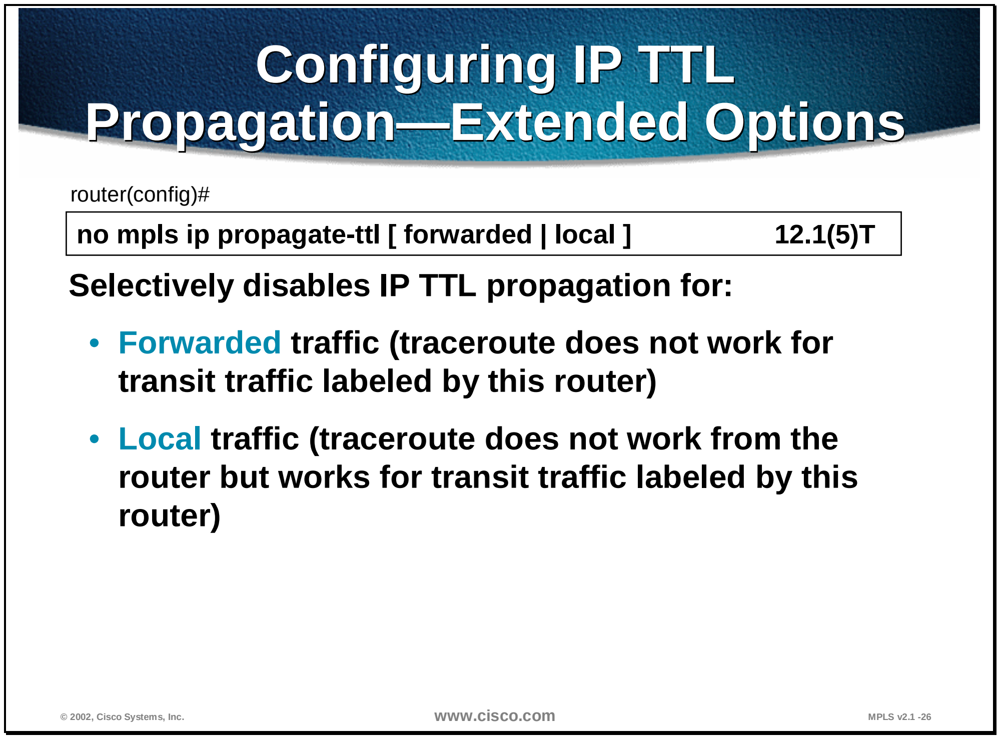
Step 1 The first packet is an IP packet with TTL=1. Router A decreases the TTL, drops the packet and sends an ICMP TTL-exceeded message to the source.

Step 2 The second packet is an IP packet with TTL=2. Router A decreases the TTL, labels the packet and sets the TTL in the MPLS header to 255.

Step 3 Router B decreases the TTL in the MPLS header to 254 and forwards a labeled packet with TTL set to 254.

Step 4 Router C removes the label, decreases the IP TTL and sends the packet to the next-hop router (C2). The packet has reached the final destination.

Note The egress MPLS router may, in some cases, be seen in the **trace** printout (for example, if the route toward C2 is carried in BGP, not in IGP).



mpls ip propagate-ttl

Use the **mpls ip propagate-ttl** command to control the generation of the time to live (TTL) field in the label when the label is first added to the IP packet. By default, this command is enabled, which means the TTL field is copied from the IP header. This allows a **traceroute** command to show all the hops in the network.

Use the **no** form of the **mpls ip propagate-ttl** command to use a fixed TTL value (255) for the first label of the IP packet. This hides the structure of the MPLS network from a **traceroute** command. The types of packets to be hidden can be specified by using the forwarded and local arguments. Specifying **no mpls ip propagate-ttl forwarded** allows the structure of the MPLS network to be hidden from customers but not the provider. This is the most common application of the command.

mpls ip propagate-ttl
no mpls ip propagate-ttl [forwarded | local]

Syntax Description

forwarded	(Optional) Hides the structure of the MPLS network from a traceroute command only for forwarded packets. Prevents the **traceroute** command from showing the hops for forwarded packets.
local	(Optional) Hides the structure of the MPLS network from a traceroute command only for local packets. Prevents the **traceroute** command from showing the hops only for local packets.

Defaults

By default, this command is enabled. The TTL field is copied from the IP header. A **traceroute** command shows all the hops in the network.

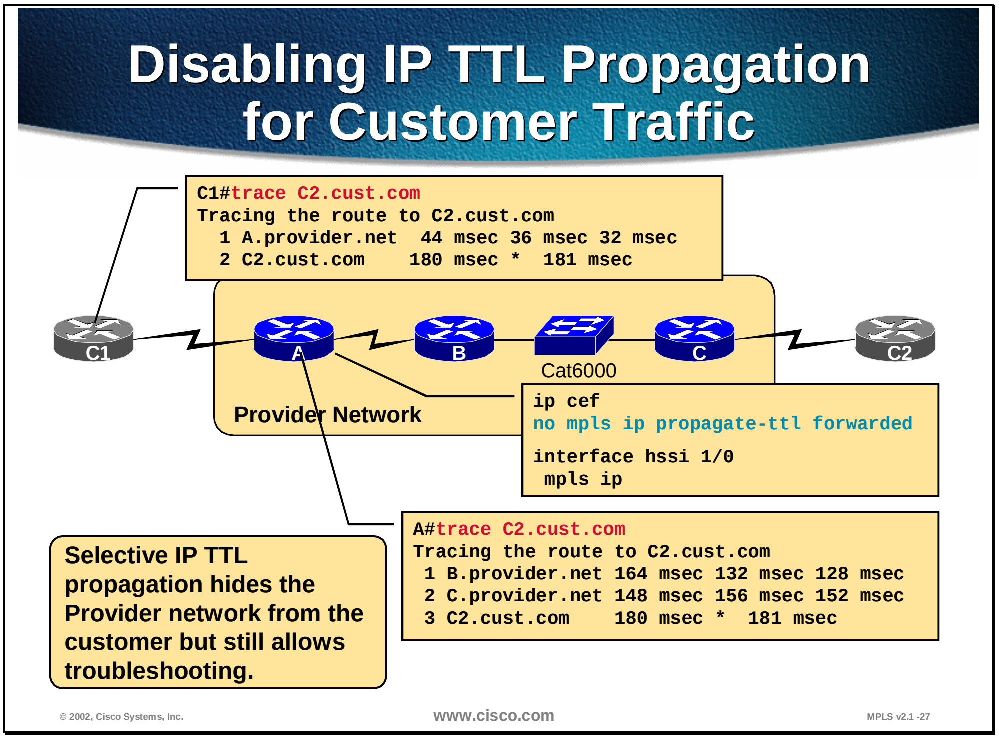
Command Modes

Global configuration

Usage Guidelines

By default, the **mpls ip propagate-ttl** command is enabled and the IP TTL value is copied to the MPLS TTL field during label imposition. To disable TTL propagation for all packets, use the **no mpls ip propagate-ttl** command. To disable TTL propagation for only forwarded packets, use the **no mpls ip propagate forward** command. This allows the structure of the MPLS network to be hidden from customers, but not the provider.

This feature supports the IETF draft documents *ICMP Extensions for MultiProtocol Label Switching*.



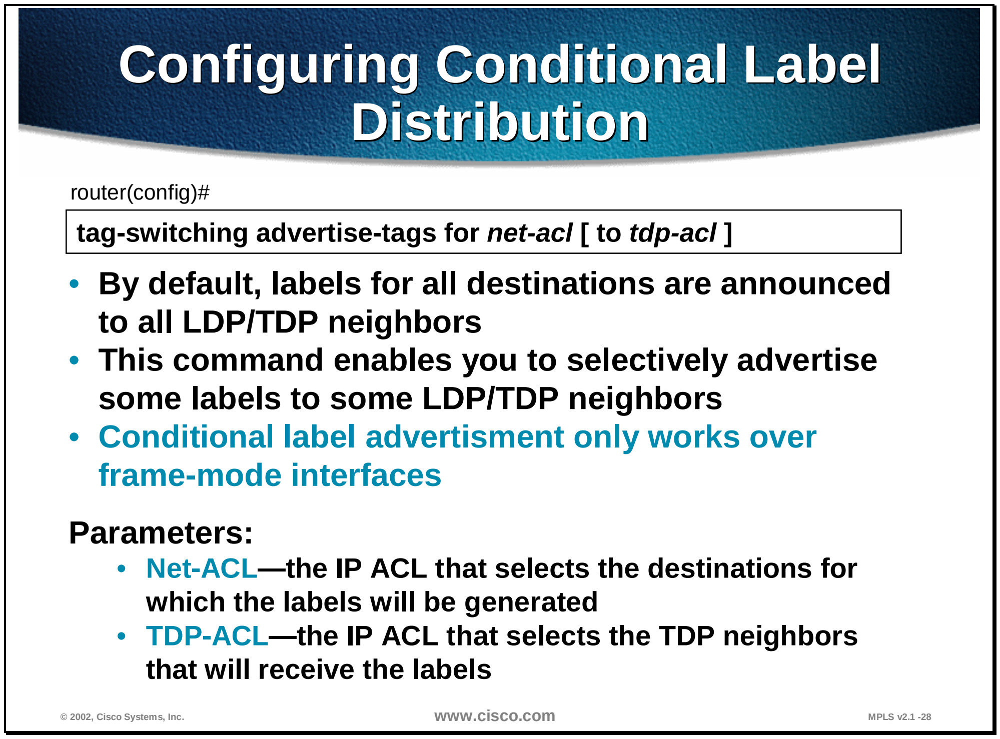
There is a third option for TTL propagation, the **no mpls ip propagate-ttl forwarded** command. Typically a service provider likes to hide the backbone from outside users but allow inside traceroute to work for easier troubleshooting of the network.

This can be achieved by disabling TTL propagation for forwarded packets only:

- If an IP packet originates in the router, the real TTL value is copied into label's TTL field
- If the IP packet is received through an interface the TTL field in a label is assigned a value of 255

The result is that someone using traceroute on a router will see all the backbone routers. Others will only see edge routers.

The opposite behavior can be achieved by using the **no mpls ip propagate-ttl local** command, although this is not usually desired.



Conditional label distribution means that a router conditionally advertises labels to neighboring routers. It will still assign local labels to all routes in the routing table that are created by an IGP. But these labels may or may not be propagated to the neighbors by the LDP or the TDP protocol.

Two access-lists are used when configuring this. The first access-list is used to select the neighbors for which the second access-list is used to filter labels.

By default all labels are distributed to all neighbors. By using the **tag-switching advertise-tags** configuration command labels for some routes in the routing table can be advertised to some neighbors while other labels are advertised to other neighbors.

tag-switching advertise-tags

To control the distribution of locally assigned (incoming) labels via the TDP, use the **tag-switching advertise-tags** command in global configuration mode. To disable label advertisement, use the **no** form of this command.

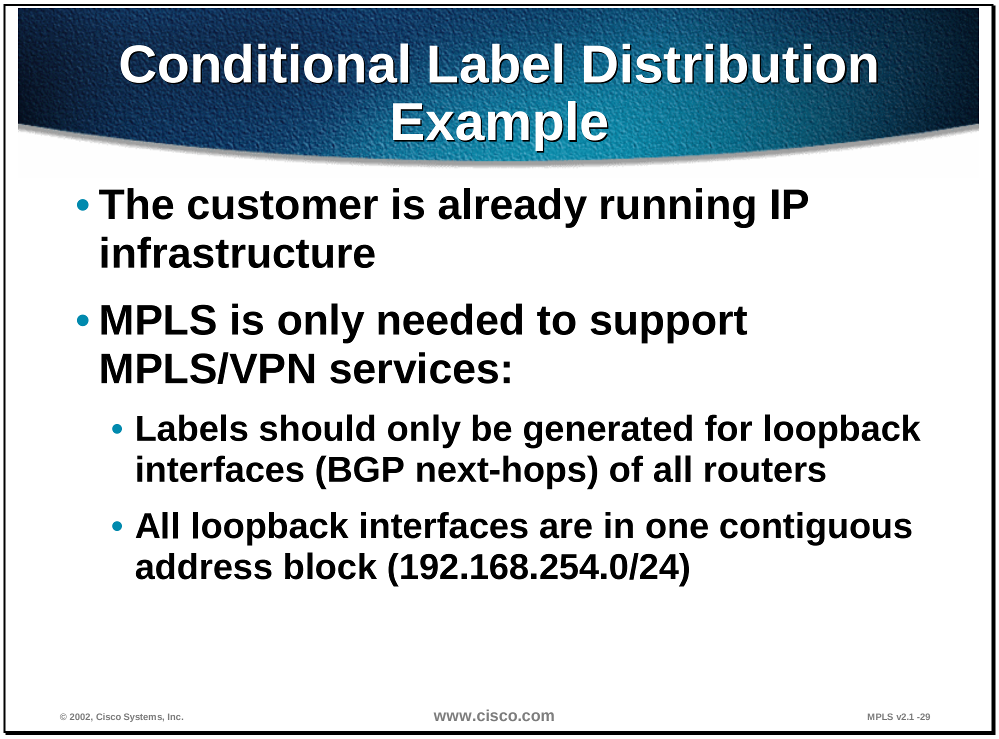
tag-switching advertise-tags [**for** *access-list-number* [**to** *access-list-number*]]
no tag-switching advertise-tags [**for** *access-list-number* [**to** *access-list-number*]]

Syntax Description

for *access-list-number*	(Optional) Specifies which destinations should have their labels advertised.
to *access-list-number*	(Optional) Specifies which LSR neighbors should receive label advertisements. The router ID that is the first 4 bytes of its 6-byte LDP identifier identifies a LSR.

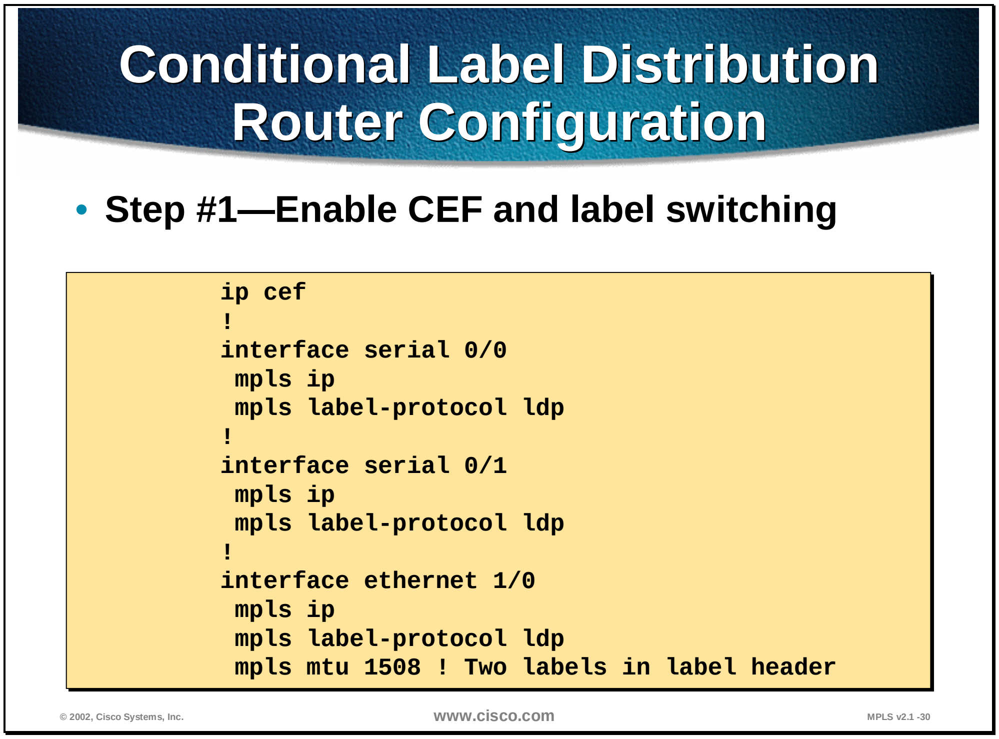
Defaults

The labels of all destinations are advertised to all LSR neighbors.



The example describes where conditional label advertising can be used. The existing network still performs normal IP routing, but the MPLS label-switched path between the loopback interfaces of the Service Provider edge routers (PE routers) is needed to enable MPLS VPN functionality.

Using one contiguous block of IP addresses for loopbacks on PE routers can simplify the configuration of conditional advertising.



In the first step CEF switching and MPLS have to be enabled on all core interfaces. LDP is used as label protocol. The MPLS MTU size may have to be adjusted on the LAN interfaces.

Conditional Label Distribution Router Configuration (Cont.)

- **Step #2—Enable conditional label advertisment**

```
!
! Disable default advertisment mechanism
!
no tag-switching advertise-tags
!
! Configure conditional advertisments
!
tag-switching advertise-tags for 90 to 91
!
access-list 90 permit ip 192.168.254.0 0.0.0.255
access-list 91 permit ip any
```

 www.cisco.com MPLS v2.1 -31

In the second step disable the label propagation and enable the conditional label advertising. Within the **tag-switching advertise-tags** command specify the neighbors to which the labels are to be sent, and the networks for which the labels are to be advertised.

In the example the labels for all networks permitted by access list 90 (which would be all routes in the 192.168.254.0/24 range) are sent to all neighbors matched by access list 91 (in the example that would be all LDP neighbors).

Summary

To enable MPLS on an interface use the **mpls ip** command or the **tag-switching ip** command on Cisco IOS software releases prior to 12.1(3)T. Use the **mpls label-protocol** command to start LDP, the standard protocol for label exchange between directly connected neighboring routers. If no label protocol is specified the Cisco proprietary protocol TDP is automatically started.

When labeled packets traverse the core network the TTL field within the label header is treated in the same way as with normal IP packets. To prevent non-edge routers from replying with ICMP *Time-to-Live Exceeded* messages disable the TTL propagation. This effectively hides the core routers from outside users. Use the **no mpls ip propagate-ttl** command to accomplish this or **no tag-switching ip propagate-ttl** on Cisco IOS software releases prior to 12.1(3)T.

Labeled packets increase in size due to the imposition of the label header and may cause problems on certain media and equipment. Use the **tag-switching mtu** command to specify the maximum size of labeled IP packets that can be transported across the interface. The value specified is used in replies to MTU discovery packets.

To prevent labeling packets for all destinations disable label propagation and enable conditional label propagation by using the **tag-switching advertise-tags** command.

Lesson Review

1. What are the mandatory configuration steps needed to enable MPLS on IOS platform?
2. How do you start LDP/TDP process in the router?
3. What are the rules for using LDP and TDP in the same LSR?
4. What is the effect of disabling TTL propagation?
5. When should you configure the MPLS mtu on an interface?
6. What happens if the MPLS mtu is not configured on an Ethernet-type interface?
7. Why would you use conditional label distribution?

Configuring Frame-Mode MPLS on Switched WAN Media

Objectives

Upon completion of this lesson, the learner will be able to perform the following tasks:

- Describe how Frame-mode MPLS runs over switched WAN media
- Identify the scenarios in which Frame-mode MPLS over switched WAN media is useful
- Understand how to configure MPLS on Frame Relay interfaces
- Explain how to configure MPLS on Frame-Mode ATM interfaces (over ATM Forum PVC)

Configuring MPLS over ATM Forum PVC

Why:

- **Run MPLS over ATM networks that do not support MPLS**
- **Potential first phase in ATM network migration**

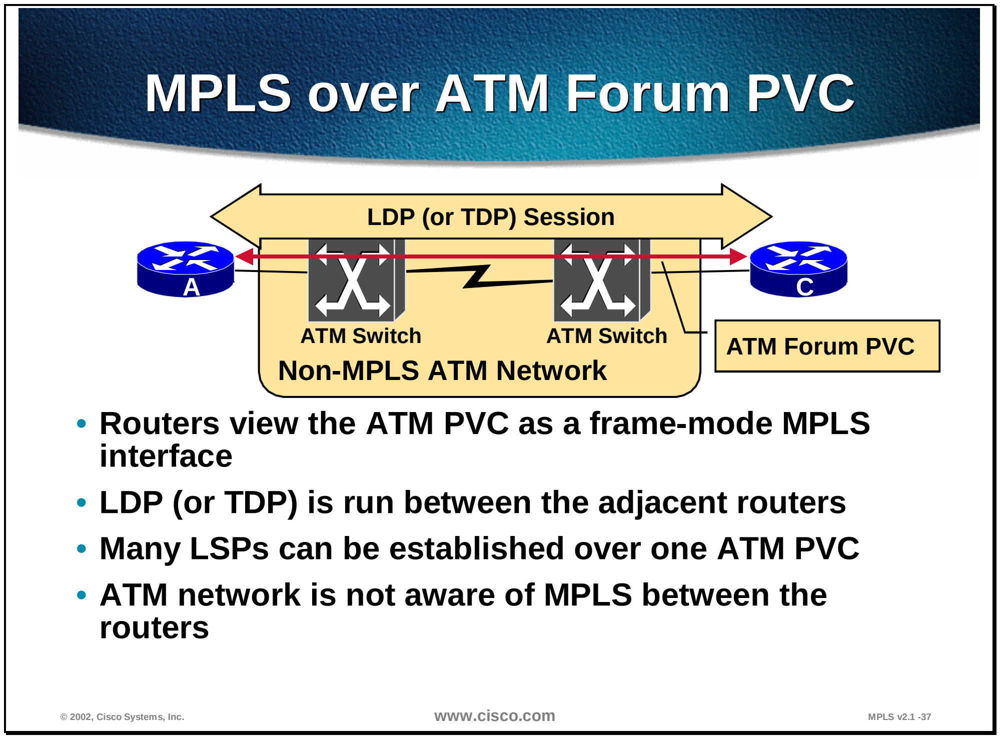
How:

- **Configure MPLS over ATM point-to-point subinterfaces on the routers**

 www.cisco.com MPLS v2.1 -36

When an underlying ATM infrastructure that does not support cell-mode MPLS is used, MPLS can still be used across point-to-point PVCs. The MPLS configuration is equal to that on any other Layer 2 media.

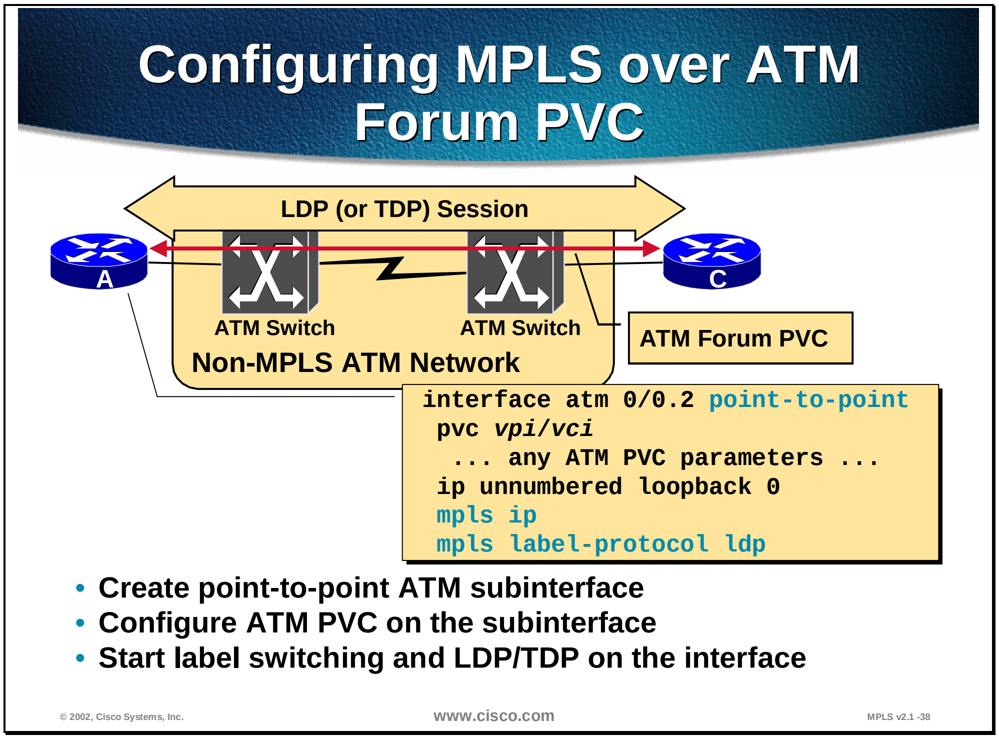
This can be the first step to introduce MPLS before the ATM network is migrated to support cell mode MPLS.



If frame-mode MPLS on an ATM interface is enabled, LDP (or TDP) neighborship is established between the two PVC endpoint routers and not with the attached ATM switch. Labeling of packets happens at the process level (in software) while segmentation and reassembly happens on the interface (in hardware), regardless of the type of packet. Switching is performed based on the VPI/VCI value in the ATM header that is used for this particular PVC and is not related to Layer 3 IP information.

From the routers' point of view, the PVC is a point-to-point connection running in frame mode MPLS. From the ATM switches' point of view, the cells exchanged between the routers belong to a single PVC. The ATM switches have no concept of MPLS in this example.

The routers may exchange several labels between them which will be used to create several LSPs across the single PVC.



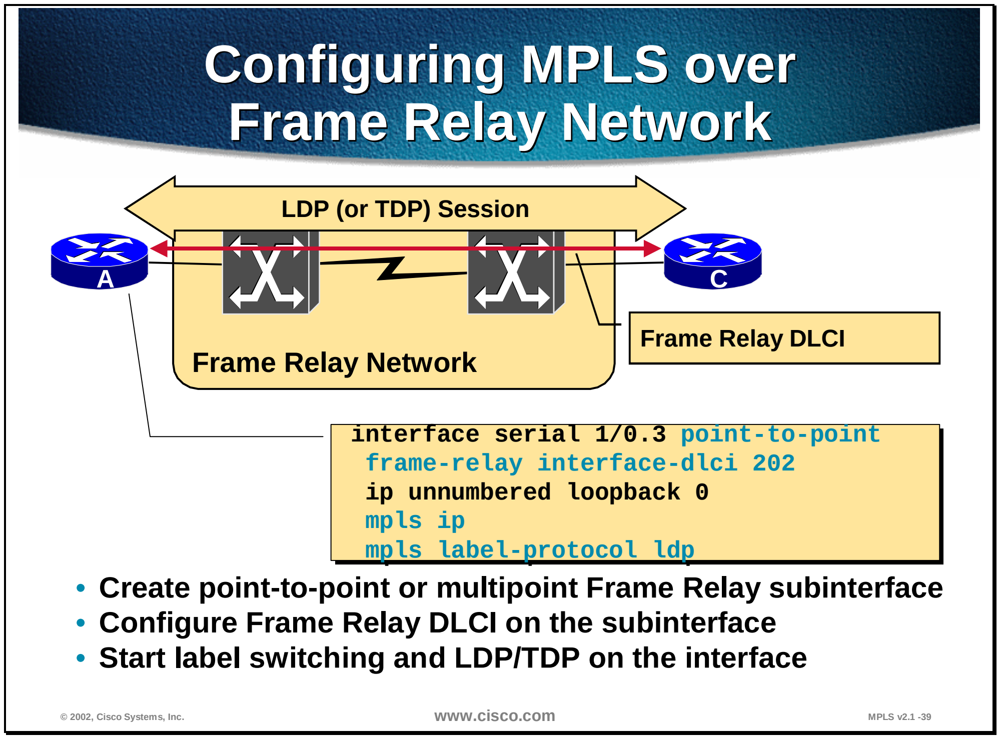
Configuring frame-mode MPLS on an ATM interface involves using the same commands as when configuring frame-mode MPLS on a leased line. The ATM parameters are not related to MPLS since the labeled traffic is using a standard ATM forum point-to-point PVC.

The steps to take are:

Step 1 Create an ATM subinterface in point-to-point mode

Step 2 Bind an ATM PVC to the subinterface.

Step 3 Start MPLS and the label protocol on the subinterface



Enabling MPLS on a Frame Relay PVC (also called a Data Link Connection Identifier [DLCI]) is no different from any other point-to-point media. Routers insert a label between the frame and the IP header. The LDP (or TDP) session is established between the two IP endpoints connected through a Frame Relay network

The steps to take are:

Step 1 Create a Frame-Relay subinterface in point-to-point mode

Step 2 Bind a Frame-Relay PVC to the subinterface.

Step 3 Start MPLS and the label protocol on the subinterface

Summary

If ATM is used within the MPLS network but the ATM switches do not support MPLS, standard PVCs can be used to interconnect routers and frame-mode MPLS can be used across the PVCs. PVC information is used to perform cell switching within the ATM network and a shim header is used to perform label switching on the routers.

Configuration tasks are equal to those on other frame-based interfaces. Use the **mpls ip** (or **tag-switching ip)** command to enable frame-mode MPLS on point-to-point (sub)interfaces.

Lesson Review

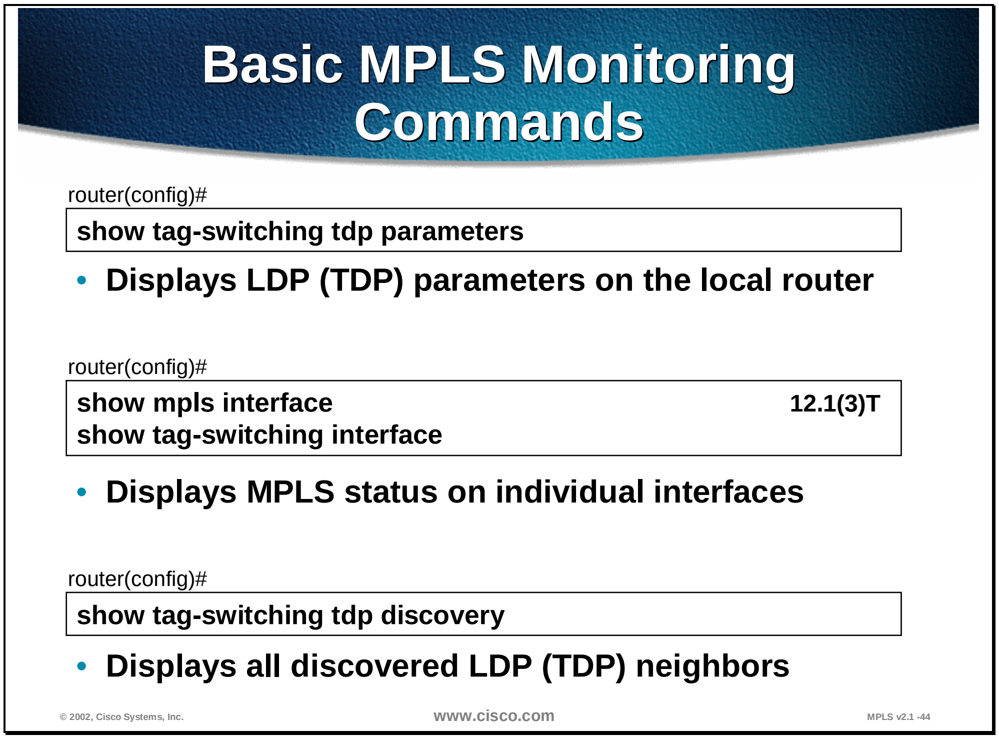
1. Why would you run frame-mode MPLS over an ATM network?
2. Does IOS support running MPLS over Frame-Relay networks?
3. List the steps needed to configure MPLS over an ATM PVC.

Monitoring MPLS on Cisco IOS—Frame-Mode Interfaces

Objectives

Upon completion of this lesson, the learner will be able to perform the following tasks:

- Describe procedures for monitoring MPLS on IOS platforms
- List the debugging commands associated with label switching, LDP and TDP
- Identify common configuration or design errors
- Use the available debugging commands in real-life troubleshooting scenarios



show tag-switching tdp parameters

To display available LDP (TDP) parameters, use the **show tag-switching tdp parameters** command in privileged EXEC mode.

show tag-switching tdp parameters

show mpls interfaces

To display information about one or more interfaces that have the MPLS feature enabled, use the **show mpls interfaces** command in EXEC mode.

show mpls interfaces [*interface*] [**detail**]

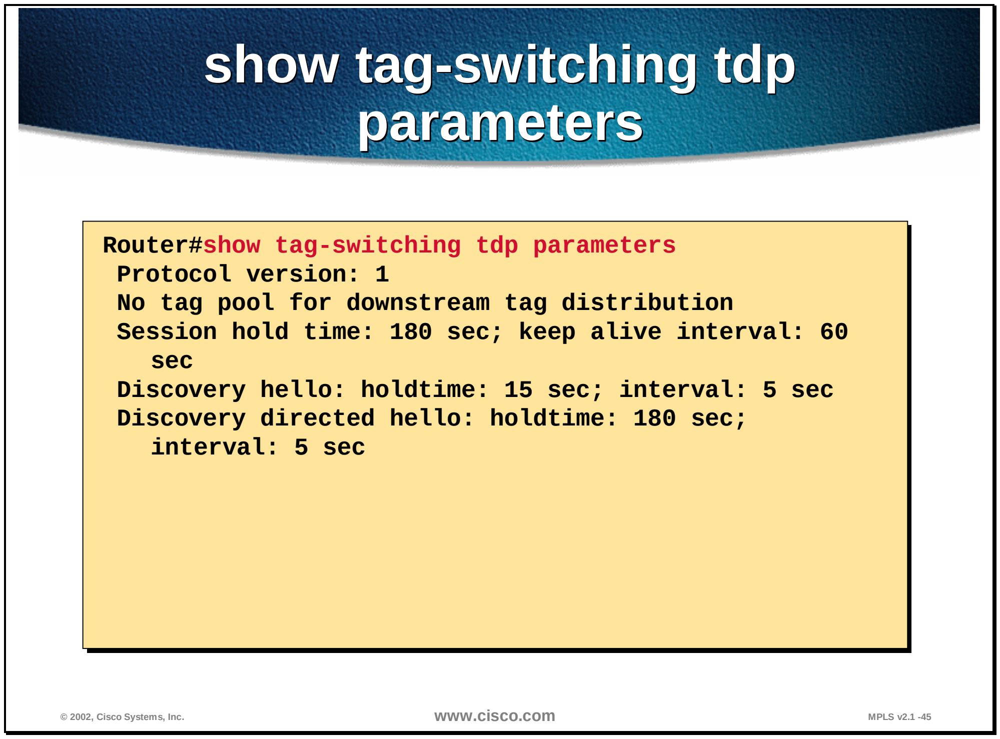
Syntax Description

interface	(Optional) The interface about which to display MPLS information.
detail	(Optional) Displays information in long form.

show tag-switching tdp discovery

To display the status of the LDP discovery process, use the **show tag-switching tdp discovery** command in privileged EXEC mode. The status of the LDP discovery process means a list of interfaces over which LDP discovery is running.

show tag-switching tdp discovery



show tag-switching tdp parameters

To display available LDP (TDP) parameters, use the **show tag-switching tdp parameters** command in privileged EXEC mode.

show tag-switching tdp parameters

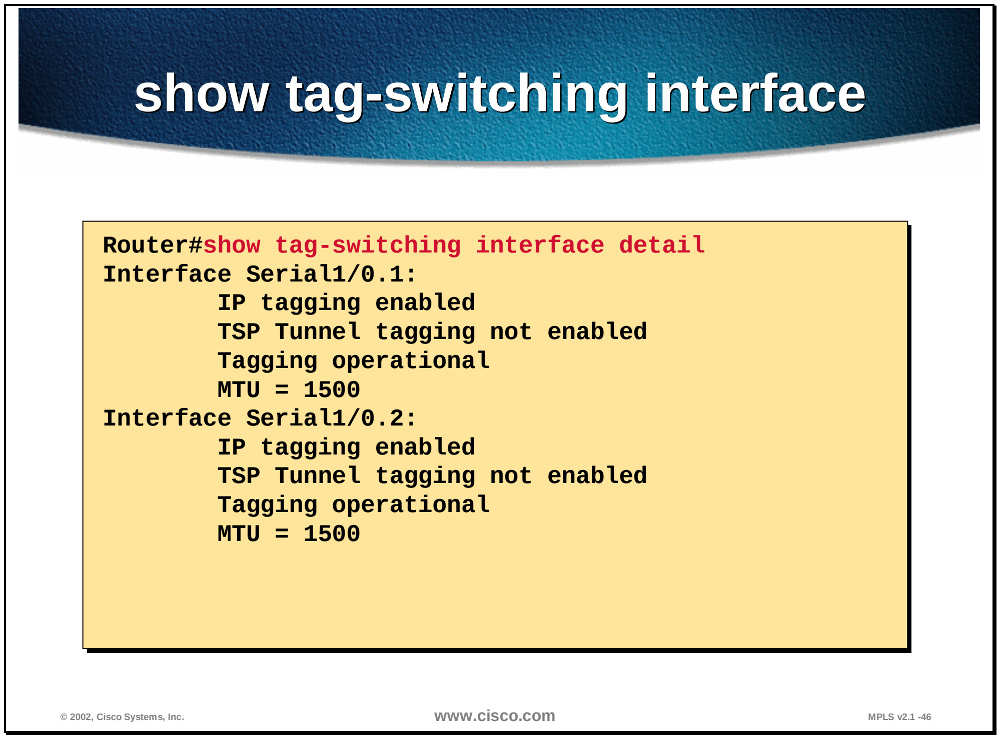
Syntax Description

This command has no arguments or keywords.

The significant fields in this display are:

Field	Description
Protocol version	Indicates the version of the LDP running on the platform.
Downstream tag pool	Describes the range of labels available for the platform to assign for Label Switching. The labels available run from the smallest label value (min label) to the largest label value (max label), with a modest number of labels at the low end of the range (reserved labels) reserved for diagnostic purposes.
Session hold time	Indicates the time to maintain a LDP session with a LDP peer device without receiving LDP traffic or a LDP keepalive from the peer device.
keep alive interval	Indicates the interval of time between consecutive transmission LDP keepalive messages to a LDP peer device.

Discovery hello	Indicates the amount of time to remember that a neighbor platform wants a LDP session without receiving a LDP Hello from the neighbor (holdtime), and the time interval between the transmitting LDP Hello messages to neighbors (interval).
Discovery directed hello	Indicates the amount of time to remember that a neighbor platform wants a LDP session when (1) the neighbor platform is not directly connected to the router and (2) the neighbor platform has not sent an LDP Hello message. The interval is known as holdtime. Also indicates the time interval between the transmission of Hello messages to a neighbor not directly connected to the router.
Accepting directed hellos	Indicates that the platform will accept and action Directed LDP Hello messages. This field may not be present.



show tag-switching interfaces

To display information about one or more interfaces that have the MPLS feature enabled, use the **show tag-switching interfaces** command in EXEC mode.

show tag-switching interfaces [*interface*] [**detail**]

Syntax Description

interface	(Optional) The interface about which to display MPLS information.
detail	(Optional) Displays information in long form.

Usage Guidelines

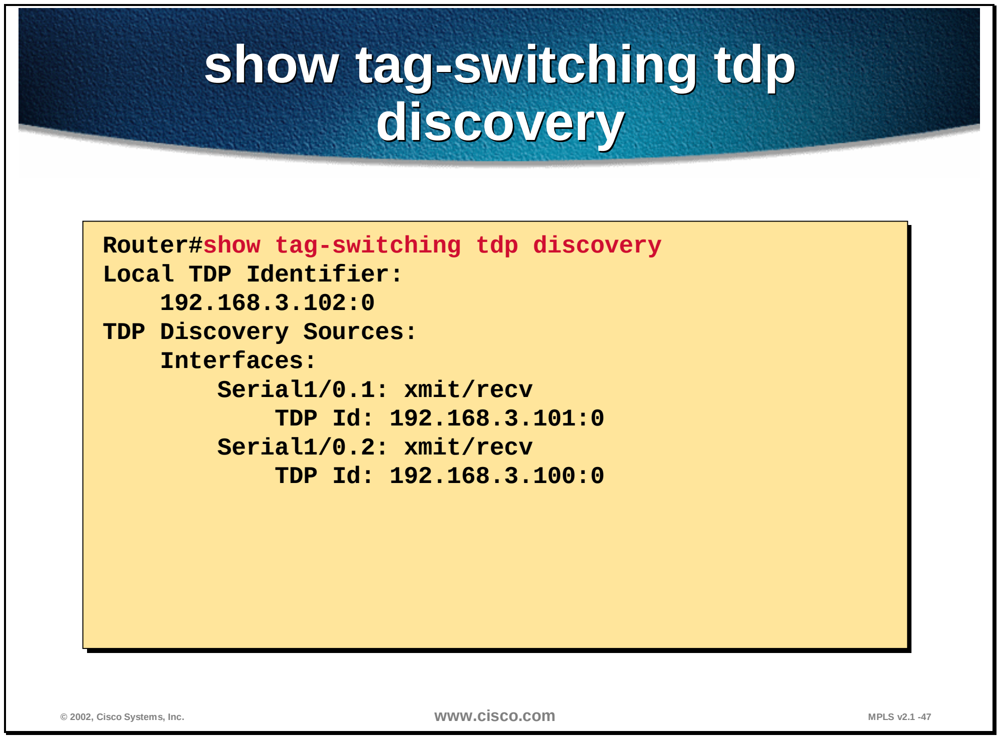
The information about the requested interface or about all interfaces on which the MPLS feature is enabled can be displayed.

The following table lists the fields displayed in this example.

Table: show tag-switching interfaces detail Field Descriptions

Field	Description
Interface	Interface type and number.
IP tagging enabled	Status of IP MPLS on an interface.
TSP Tunnel tagging not enabled	Status of label lsp-tunnels on the interface.
Tagging operational	Operational status of MPLS on an interface.
Tagswitching feature vector	Specifies the MPLS feature vector on an interface.
MTU	Maximum number of data bytes per labeled packet that will be transmitted.

ATM tagging	The interface uses TC-ATM procedures.



show tag-switching tdp discovery

To display the status of the LDP discovery process, use the **show tag-switching tdp discovery** command in privileged EXEC mode. The status of the LDP discovery process means a list of interfaces over which LDP discovery is running.

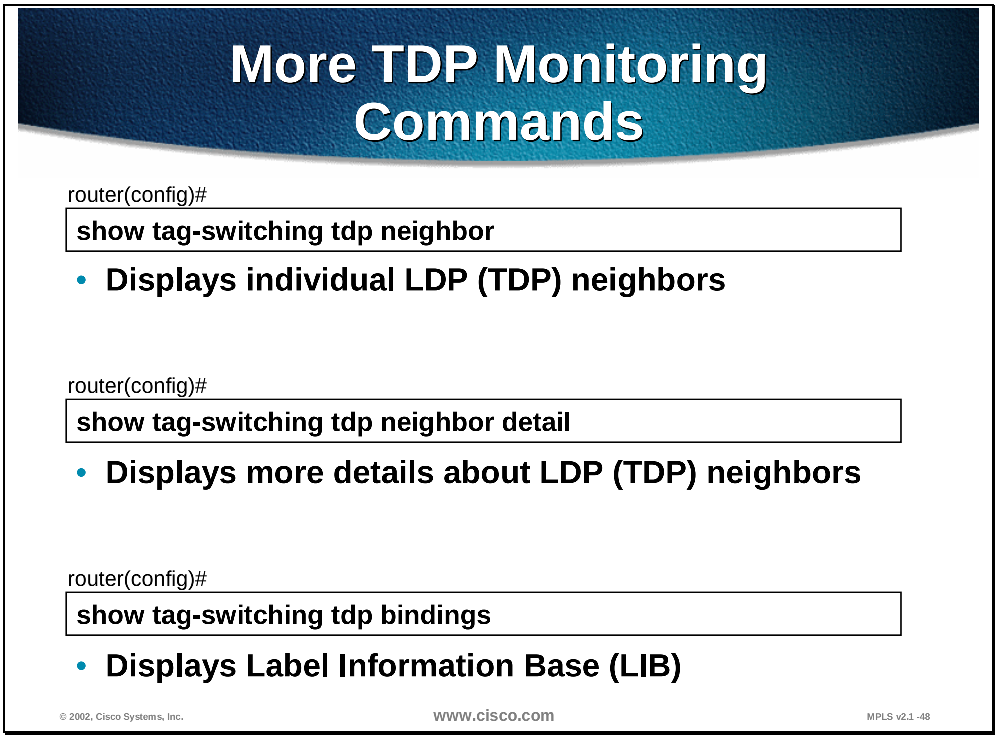
show tag-switching tdp discovery

Syntax Description

This command has no arguments or keywords.

The following table describes the significant fields in this display.

Field	Description
Local TDP Identifier	The LDP identifier for the local router. A LDP identifier is a 6-byte quantity displayed as an IP address:number. The Cisco convention is to use a router ID for the first 4 bytes of the LDP identifier and integers starting with 0 for the final two bytes of the IP address:number.
Interfaces	Lists the interfaces engaging in LDP discovery activity. xmit indicates that the interface is transmitting LDP discovery hello packets; recv indicates that the interface is receiving LDP discovery hello packets.



show tag-switching tdp neighbors

To display the status of LDP sessions, use the **show tag-switching tdp neighbors** command in privileged EXEC mode.

show tag-switching tdp neighbors [*address* | *interface*] [**detail**]

Syntax Description

address	(Optional) The neighbor that has this IP address.
interface	(Optional) LDP neighbors accessible over this interface.
detail	(Optional) Displays information in long form.

show tag-switching tdp bindings

To display the contents of the label information base (LIB), use the **show tag-switching tdp bindings** command in privileged EXEC mode.

show tag-switching tdp bindings [*network* {*mask* | *length*} [**longer-prefixes**]] [**local-tag** *tag* [**-** *tag*]] [**remote-tag** *tag* [**-** *tag*]] [**neighbor** *address*] [**local**]

Syntax Description

network	(Optional) Destination network number.
mask	(Optional) Network mask written as A.B.C.D.
length	(Optional) Mask length (1 to 32 characters).
longer-prefixes	(Optional) Selects any prefix that matches *mask* with *length* to 32.

local-tag *tag* - *tag*	(Optional) Displays entries matching local label values by this router. Use the **-** *tag* argument to indicate label range.
remote-tag *tag* - *tag*	(Optional) Displays entries matching label values assigned by a neighbor router. Use the **-** *tag* argument to indicate label range.
neighbor *address*	(Optional) Displays label bindings assigned by selected neighbor.
local	(Optional) Displays local label bindings.

show tag tdp neighbor

```
Router#show tag-switching tdp neighbors
Peer TDP Ident: 192.168.3.100:0; Local TDP Ident
192.168.3.102:0
        TCP connection: 192.168.3.100.711 - 192.168.3.102.11000
        State: Oper; PIEs sent/rcvd: 55/53; ; Downstream
        Up time: 00:43:26
        TDP discovery sources:
          Serial1/0.2
        Addresses bound to peer TDP Ident:
          192.168.3.10    192.168.3.14    192.168.3.100
```


show tag-switching tdp neighbors

To display the status of LDP sessions, use the **show tag-switching tdp neighbors** command in privileged EXEC mode.

show tag-switching tdp neighbors [*address* | *interface*] [**detail**]

Usage Guidelines

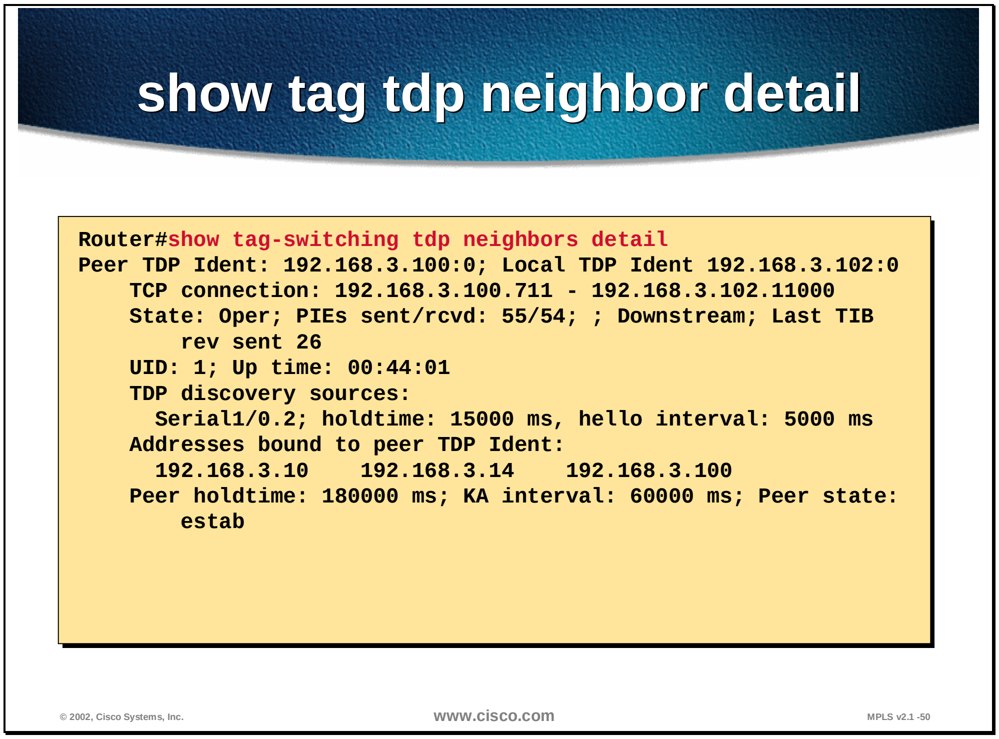
The neighbor information branch can give information about all LDP neighbors, or it can be limited to:

- The neighbor with a specific IP address
- LDP neighbors known to be accessible over a specific interface

The significant fields in this display are:

Field	Description
Peer TDP Ident	The LDP identifier of the neighbor (peer device) for this session.
Local TDP Ident	The LDP identifier for the local LSR (TSR) for this session.
TCP connection	The TCP connection used to support the LDP session. The format for displaying the TCP connection is: *peer IP address.peer port* *local IP address.local port*

State	The state of the LDP session. Generally this is Oper (operational), but Transient is another possible state.
PIEs sent/rcvd	The number of LDP protocol information elements (PIEs) sent to, and received from, the session peer device. The count includes the transmission and receipt of periodic keepalive PIEs which are required for maintenance of the LDP session.
Downstream	Indicates that the downstream method of label distribution is being used for this LDP session. When the downstream method is used, a LSR advertises all of its locally assigned (incoming) labels to its LDP peer device (subject to any configured access list restrictions).
Downstream on demand	Indicates that the downstream-on-demand method of label distribution is being used for this LDP session. When the downstream-on-demand method is used, a LSR advertises its locally assigned (incoming) labels to its LDP peer device only when the peer device asks for them.
Up time	The length of time the LDP session has existed.
TDP discovery sources	The source(s) of LDP discovery activity that led to the establishment of this LDP session.
Addresses bound to peer TDP Ident	The known interface addresses of the LDP session peer device. These are addresses that may appear as next-hop addresses in the local routing table. They are used to maintain the Label Forwarding Information Base (LFIB).



show tag-switching tdp neighbors detail

To display the detailed status of LDP sessions, use the **show tag-switching tdp neighbors detail** command in privileged EXEC mode.

show tag-switching tdp neighbors [*address* | *interface*] **detail**

Usage Guidelines

The neighbor information branch can give information about all LDP neighbors, or it can be limited to:

- The neighbor with a specific IP address
- LDP neighbors known to be accessible over a specific interface

The significant fields in this display are:

Field	Description
Peer TDP Ident	The LDP identifier of the neighbor (peer device) for this session.
Local TDP Ident	The LDP identifier for the local LSR (TSR) for this session.
TCP connection	The TCP connection used to support the LDP session. The format for displaying the TCP connection is: *peer IP address.peer port* *local IP address.local port*

State	The state of the LDP session. Generally this is Oper (operational), but Transient is another possible state.
PIEs sent/rcvd	The number of LDP protocol information elements (PIEs) sent to, and received from, the session peer device. The count includes the transmission and receipt of periodic keepalive PIEs, which are required for maintenance of the LDP session.
Downstream	Indicates that the downstream method of label distribution is being used for this LDP session. When the downstream method is used, a LSR advertises all of its locally assigned (incoming) labels to its LDP peer device (subject to any configured access list restrictions).
Downstream on demand	Indicates that the downstream-on-demand method of label distribution is being used for this LDP session. When the downstream-on-demand method is used, a LSR advertises its locally assigned (incoming) labels to its LDP peer device only when the peer device asks for them.
Up time	The length of time the LDP session has existed.
TDP discovery sources	The source(s) of LDP discovery activity that led to the establishment of this LDP session. It also displays the hold-time and hello interval used with discovery messages.
Addresses bound to peer TDP Ident	The known interface addresses of the LDP session peer device. These are addresses that may appear as next-hop addresses in the local routing table. They are used to maintain the LFIB.
Peer holdtime	Displays the time it takes to remove the neighborship if no keepalives are received within this period.
KA interval	Displays the keepalive interval.
Peer state	Shows the status of the neighborship

show tag tdp bindings

```
Router#show tag tdp bindings
  tib entry: 192.168.3.1/32, rev 9
     local binding:  tag: 28
     remote binding: tsr: 19.16.3.3:0, tag: 28
  tib entry: 192.168.3.2/32, rev 8
     local binding:  tag: 27
     remote binding: tsr: 19.16.3.3:0, tag: 27
  tib entry: 192.168.3.3/32, rev 7
     local binding:  tag: 26
     remote binding: tsr: 19.16.3.3:0, tag: imp-null(1)
  tib entry: 192.168.3.10/32, rev 6
     local binding:  tag: imp-null(1)
     remote binding: tsr: 19.16.3.3:0, tag: 26
```

 www.cisco.com MPLS v2.1 -51

show tag-switching tdp bindings

To display the contents of the label information base (LIB), use the **show tag-switching tdp bindings** command in privileged EXEC mode.

show tag-switching tdp bindings [*network* {*mask* | *length*} [**longer-prefixes**]] [**local-tag** *tag* [**-** *tag*]] [**remote-tag** *tag* [**-** *tag*]] [**neighbor** *address*] [**local**]

Syntax Description

network	(Optional) Destination network number.
mask	(Optional) Network mask written as A.B.C.D.
length	(Optional) Mask length (1 to 32 characters).
longer-prefixes	(Optional) Selects any prefix that matches *mask* with *length* to 32.
local-tag *tag* - *tag*	(Optional) Displays entries matching local label values by this router. Use the - *tag* argument to indicate label range.
remote-tag *tag* - *tag*	(Optional) Displays entries matching label values assigned by a neighbor router. Use the - *tag* argument to indicate label range.
neighbor *address*	(Optional) Displays label bindings assigned by selected neighbor.
local	(Optional) Displays local label bindings.

Usage Guidelines

A request can specify that the entire database be shown, or it or can be limited to a subset of entries. A request to show a subset of entries can be based on the prefix, on input or output label values or ranges, and/or on the neighbor advertising the label.

Examples

The following is sample output from the **show tag-switching tdp bindings** command. This form of the command causes the contents of the entire LIB (TIB) to be displayed.

```
show tag-switching tdp bindings

Matching entries:
  tib entry: 10.92.0.0/16, rev 28
        local binding:  tag: imp-null(1)
        remote binding: tsr: 172.27.32.29:0, tag: imp-null(1)
  tib entry: 10.102.0.0/16, rev 29
        local binding:  tag: 26
        remote binding: tsr: 172.27.32.29:0, tag: 26
  tib entry: 10.105.0.0/16, rev 30
        local binding:  tag: imp-null(1)
        remote binding: tsr: 172.27.32.29:0, tag: imp-null(1)
  tib entry: 10.205.0.0/16, rev 31
        local binding:  tag: imp-null(1)
        remote binding: tsr: 172.27.32.29:0, tag: imp-null(1)
  tib entry: 10.211.0.7/32, rev 32
        local binding:  tag: 27
        remote binding: tsr: 172.27.32.29:0, tag: 28
  tib entry: 10.220.0.7/32, rev 33
        local binding:  tag: 28
        remote binding: tsr: 172.27.32.29:0, tag: 29
  tib entry: 99.101.0.0/16, rev 35
        local binding:  tag: imp-null(1)
        remote binding: tsr: 172.27.32.29:0, tag: imp-null(1)
  tib entry: 100.101.0.0/16, rev 36
        local binding:  tag: 29
        remote binding: tsr: 172.27.32.29:0, tag: imp-null(1)
  tib entry: 171.69.204.0/24, rev 37
        local binding:  tag: imp-null(1)
        remote binding: tsr: 172.27.32.29:0, tag: imp-null(1)
  tib entry: 172.27.32.0/22, rev 38
        local binding:  tag: imp-null(1)
        remote binding: tsr: 172.27.32.29:0, tag: imp-null(1)
  tib entry: 210.10.0.0/16, rev 39
        local binding:  tag: imp-null(1)
  tib entry: 210.10.0.8/32, rev 40
        remote binding: tsr: 172.27.32.29:0, tag: 27
```

The following is a sample output from the **show tag tdp bindings 10.0.0.0 8 longer-prefixes neighbor 172.27.32.29** variant of the command; it displays labels learned from LSR (TSR) 172.27.32.29 for network 10.0.0.0 and any of its subnets. The use of the **neighbor** option suppresses the output of local labels and labels learned from other neighbors.

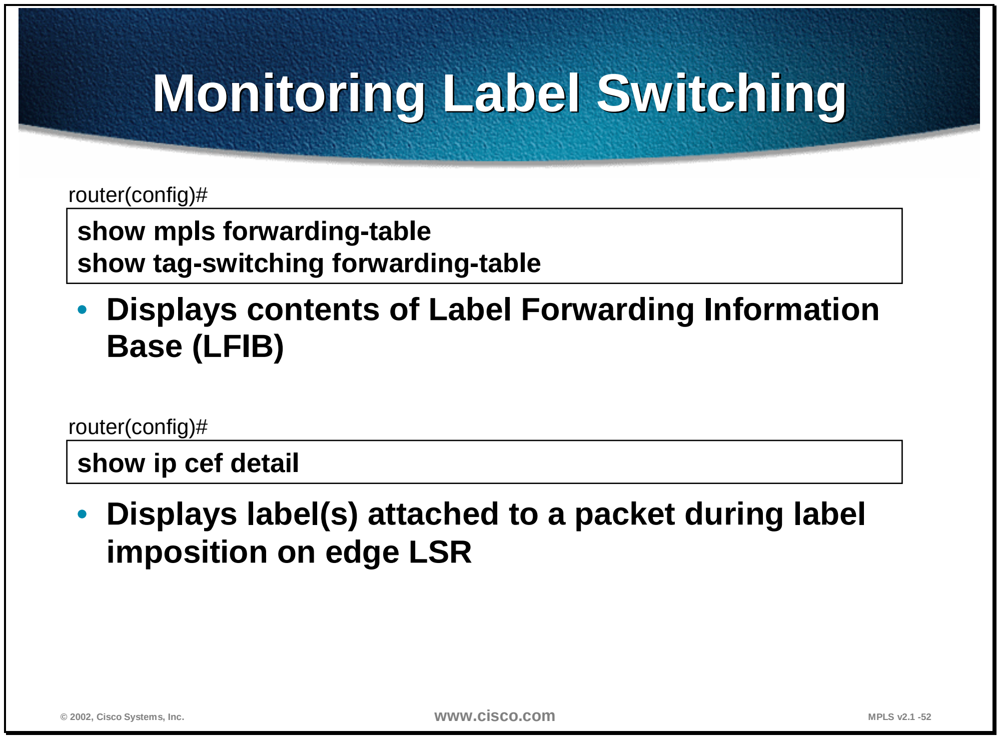
```
show tag tdp bindings 10.0.0.0 8 longer-prefixes neighbor
172.27.32.29

  tib entry: 10.92.0.0/16, rev 28
        remote binding: tsr: 172.27.32.29:0, tag: imp-null(1)
  tib entry: 10.102.0.0/16, rev 29
        remote binding: tsr: 172.27.32.29:0, tag: 26
  tib entry: 10.105.0.0/16, rev 30
        remote binding: tsr: 172.27.32.29:0, tag: imp-null(1)
  tib entry: 10.205.0.0/16, rev 31
        remote binding: tsr: 172.27.32.29:0, tag: imp-null(1)
  tib entry: 10.211.0.7/32, rev 32
```

```
        remote binding: tsr: 172.27.32.29:0, tag: 28
  tib entry: 10.220.0.7/32, rev 33
        remote binding: tsr: 172.27.32.29:0, tag: 29
```

The significant fields in this display are:

Field	Description
tib entry	Indicates that the following lines are the LIB (TIB) entry for a particular destination (network/mask). The revision number is used internally to manage label distribution for this destination.
remote binding	A list of outgoing labels for this destination learned from other Label Switching Routers (LSRs). Each item on this list identifies the LSR from which the outgoing label was learned and the label itself. Its LDP identifier identifies the LSR.
imp-null	The implicit null label. This label value instructs the upstream router to pop the label entry off the label stack before forwarding the packet.



show mpls forwarding-table

To display the contents of the Label Forwarding Information Base (LFIB), use the **show mpls forwarding-table** command in privileged EXEC mode.

show mpls forwarding-table [{*network* {*mask* | *length*} | **tags** *tag* [**-** *tag*] | **interface** *interface* | **next-hop** *address* | **tsp-tunnel** [*tunnel-id*]}] [**detail**]

show ip cef

To display entries in the FIB that are unresolved, or to display a summary of the FIB, use this form of the **show ip cef** command in EXEC mode:

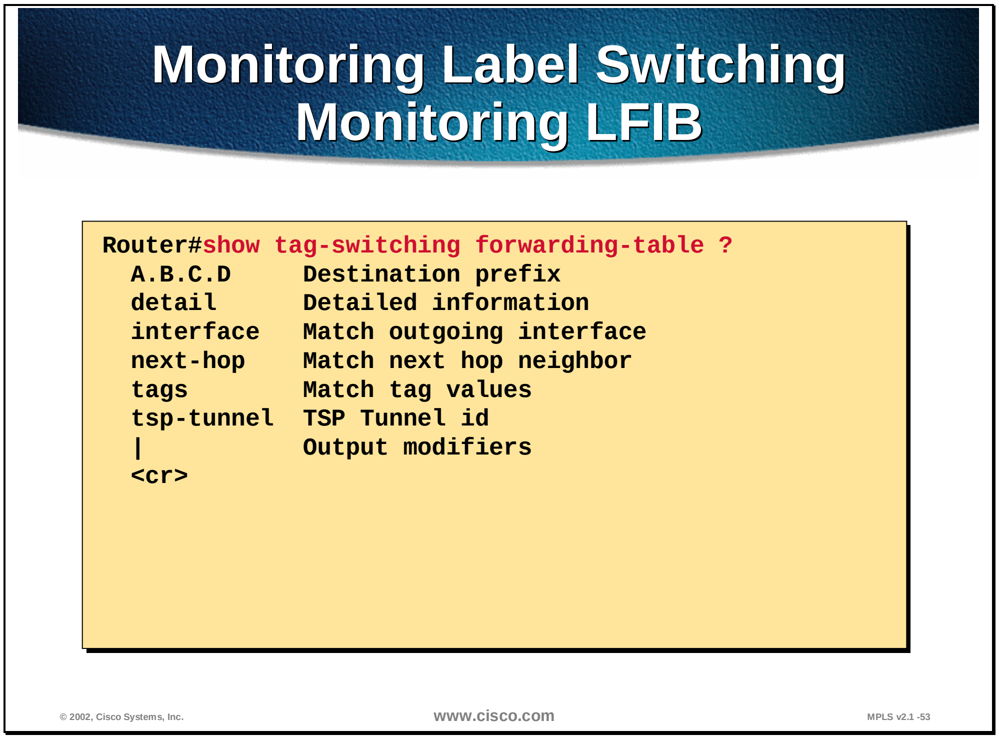
show ip cef [**unresolved** | **summary**]

To display specific entries in the FIB based on IP address information, use this form of the show ip cef EXEC command:

show ip cef [*network* [*mask* [**longer-prefix**]]] [**detail**]

To display specific entries in the FIB based on interface information, use this form of the show ip cef EXEC command:

show ip cef [*type number*] [**detail**]



show tag-switching forwarding-table

To display the contents of the Tag Forwarding Information Base (TFIB), use the **show tag-switching forwarding-table** command in the privileged EXEC command mode.

show tag-switching forwarding-table [{network {mask | length} | tags tag [-tag] | interface interface | next-hop address | tsp-tunnel [tunnel-id]}] [detail]

Syntax Description

network	(Optional.) Shows entry for specified destination only.
mask	IP address of destination mask whose entry is to be shown.
length	Number of bits in mask of destination.
tag *tag* **-** *tag*	(Optional.) Shows entries with specified local tags only.
interface *interface*	(Optional.) Shows entries with specified outgoing interface only.
next-hop *address*	(Optional.) Shows entries with specified neighbor as next hop only.
tsp-tunnel ***[tunnel-id]***	(Optional.) Shows entries with specified TSP tunnel only, or all TSP tunnel entries.
detail	(Optional.) Displays information in long form (includes length of encapsulation, length of the MAC string, maximum transmission unit (MTU), and all tags).

show tag-switching forwarding-table

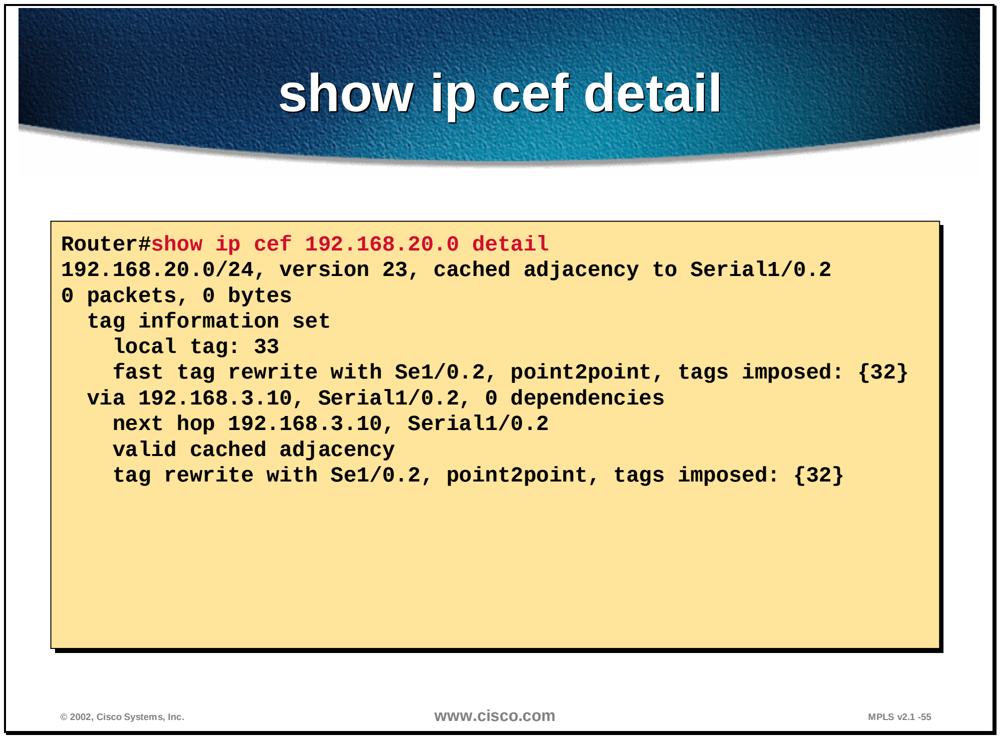
```
Router#show tag-switching forwarding-table detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     Untagged    192.168.3.3/32    0          Se1/0.3    point2point
      MAC/Encaps=0/0, MTU=1504, Tag Stack{}
27     Pop tag     192.168.3.4/32    0          Se0/0.4    point2point
      MAC/Encaps=4/4, MTU=1504, Tag Stack{}
      20618847
28     29          192.168.3.4/32    0          Se1/0.3    point2point
      MAC/Encaps=4/8, MTU=1500, Tag Stack{29}
      18718847 0001D000
```

 www.cisco.com MPLS v2.1 -54

show tag-switching forwarding-table Detail Field Descriptions

Local tag	Tag assigned by this router.
Outgoing tag or VC	Tag assigned by next hop, or VPI/VCI used to get to the next hop. Some of the entries that can be in this column are: – [T] means forwarding through a TSP tunnel. – "Untagged" means there is no tag for the destination from the next hop, or Tag Switching is not enabled on the outgoing interface. – "Pop tag" means the next hop advertised an implicit NULL tag for the destination, and this router popped the top tag.
Prefix or Tunnel Id	Address or tunnel to which the packets with this tag are going.
Bytes tag switched	Number of bytes switched with this incoming tag.
Outgoing interface	Interface through which the packets with this tag are sent.
NextHop	IP address of the neighbor that assigned the outgoing tag.
Mac/Encaps	Length in bytes of the Layer 2 header, and length in bytes of packet encapsulation, including the Layer 2 header and the tag header.
MTU	Maximum transmission unit (MTU) of tagged packet.

Tag Stack	All the outgoing tags. If the outgoing interface is TC-ATM, the VCD is also shown.
00020900 00002000	The actual encapsulation in hexadecimal form. There is a space shown between Layer 2 and the tag header.



show ip cef detail

To display the requested entries from the CEF FIB table, use the **show ip cef detail** EXEC command.

show ip cef [network [mask [**longer-prefix**]]] [**detail**]

Syntax Description

unresolved	(Optional) Displays unresolved FIB entries.
summary	(Optional) Displays a summary of the FIB.
network	(Optional) Displays the FIB entry for the specified destination network.
mask	(Optional) Displays the FIB entry for the specified destination network and mask.
longer-prefix	(Optional) Displays FIB entries for all more specific destinations.
detail	(Optional) Displays detailed FIB entry information.

Usage Guidelines

This command first appeared to support the Cisco 12012 Gigabit Switch Router in Cisco IOS Release 11.2 GS, and first appeared with multiple platform support in Cisco IOS Release 11.1 CC.

The **show ip cef** command without any keywords or arguments shows a brief display of all FIB entries.

The **show ip cef detail** command shows detailed FIB entry information for all FIB entries.

Debugging Label Switching and TDP

router(config)#

debug tag-switching tdp ...

- **Debugs TDP adjacencies, session establishment, and label bindings exchange**

router(config)#

debug mpls lfib … **12.1(3)T**
debug tag-switching tfib ...

- **Debugs Label Forwarding Information Base events: label creations, removals, rewrites**

router(config)#

debug mpls packets [*interface*] **12.1(3)T**
debug tag-switching packets [interface]

- **Debugs labeled packets switched by the router**
- **Disables fast or distributed tag switching**

 www.cisco.com MPLS v2.1 -56

There is a large number of debug commands associated with label switching on Cisco IOS. The **debug tag-switching tdp** set of commands debug various aspects of TDP protocol, from label distribution to exchange of the application-layer data between adjacent TDP-speaking routers.

The **debug mpls lfib** and equivalent **debug tag-switching tfib** set of commands display LFIB-related events (allocation of new labels, removal of labels etc.)

The **debug mpls packets** and equivalent **debug tag-switching packets** commands display all labeled packets switched by the router (through the specified interface).

Note Use this command with care because it generates output for every packet processed. Furthermore, enabling this command causes fast and distributed label switching to be disabled for the selected interfaces. To avoid adversely affecting other system activity, use this command only when traffic on the network is at a minimum.

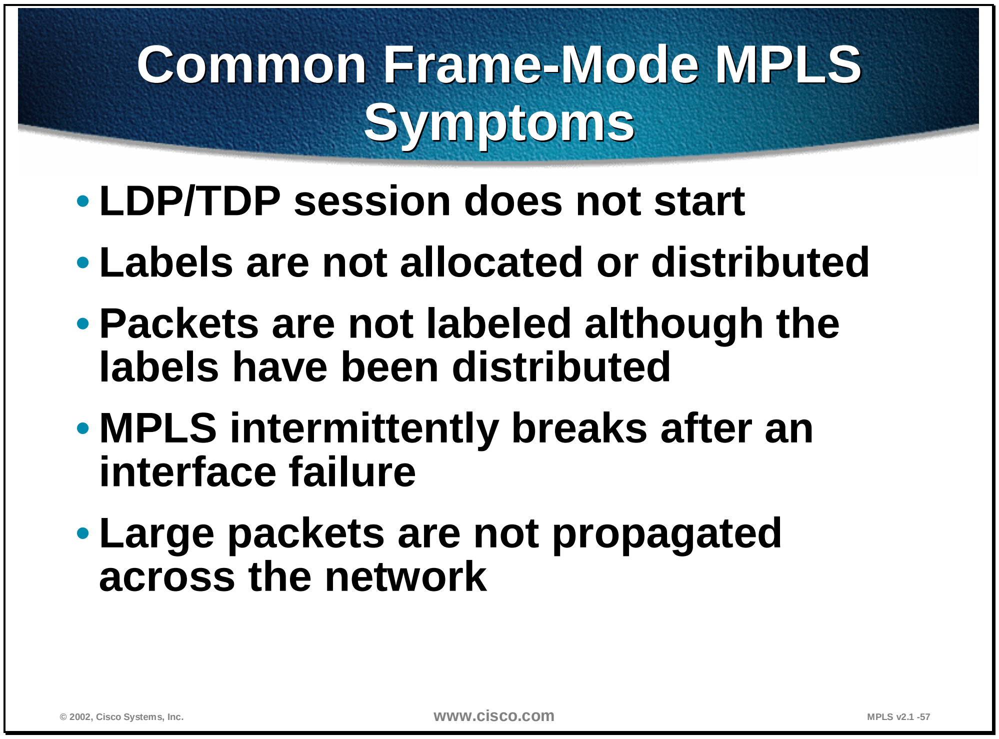
debug mpls packets
debug tag-switching packets

Use the **debug mpls packets** or **debug tag-switchings packets** EXEC command to display tagged packets switched by this router. The **no** form of this command disables debugging output.

[**no**] **debug mpls packets** [*interface*]
[**no**] **debug tag-switching packets** [*interface*]

Syntax

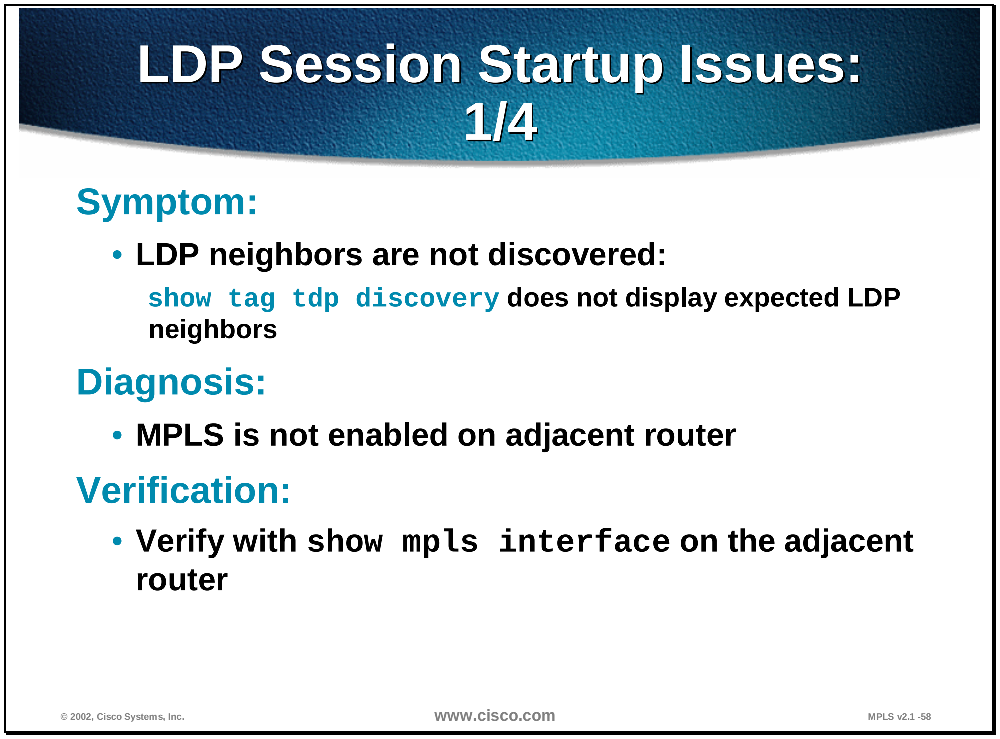
interface	(Optional) Interface or subinterface name.



There are a number of common symptoms that can be encountered while troubleshooting a Frame-mode MPLS network:

- The LDP (or TDP) session does not start
- LDP (or TDP) session starts but the labels are not allocated or distributed
- Labels are allocated and distributed, but the forwarded packets are not labeled
- MPLS stops working intermittently after an interface failure, even on interfaces totally unrelated to the failed interface
- Large IP packets are not propagated across MPLS backbone even though they were successfully propagated across pure IP backbone.

Each of these symptoms will be described and detailed information about troubleshooting provided.



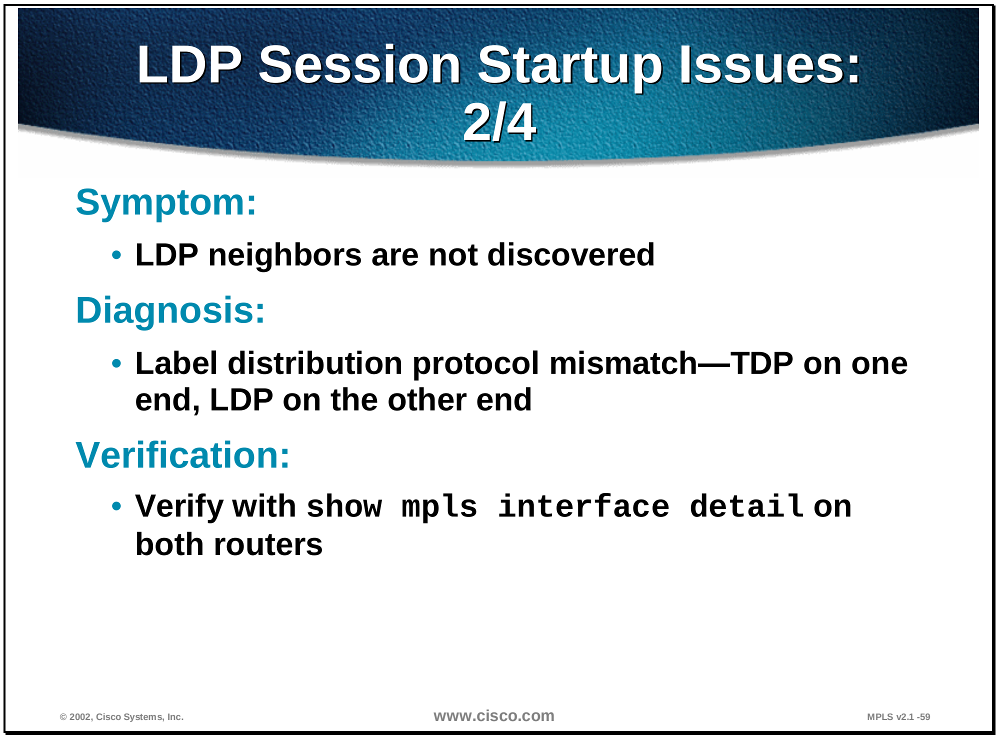
LDP (or TDP) Session Startup Problem

Symptom: LDP neighbors are not discovered. The **show tag-switching tdp discovery** command does not display the expected LDP (or TDP) neighbor.

Diagnose: If MPLS is enabled on an interface but no neighbors are discovered, it is likely that MPLS is not enabled on the neighbor. The router is sending discovery messages but the neighbor is not replying because it does not have LDP enabled.

Verification: Use the **show mpls interface** command on the adjacent router to verify if MPLS is enabled on the interface.

Solution: Enable MPLS on the neighboring router.



LDP (or TDP) Session Startup Problem

Symptom: LDP neighbors are not discovered.

Diagnose: Another possibility is that the neighbor has a different label distribution protocol enabled on the interface.

Verification: Use the **show mpls interface detail** command on both routers to verify which label protocol that is enabled.

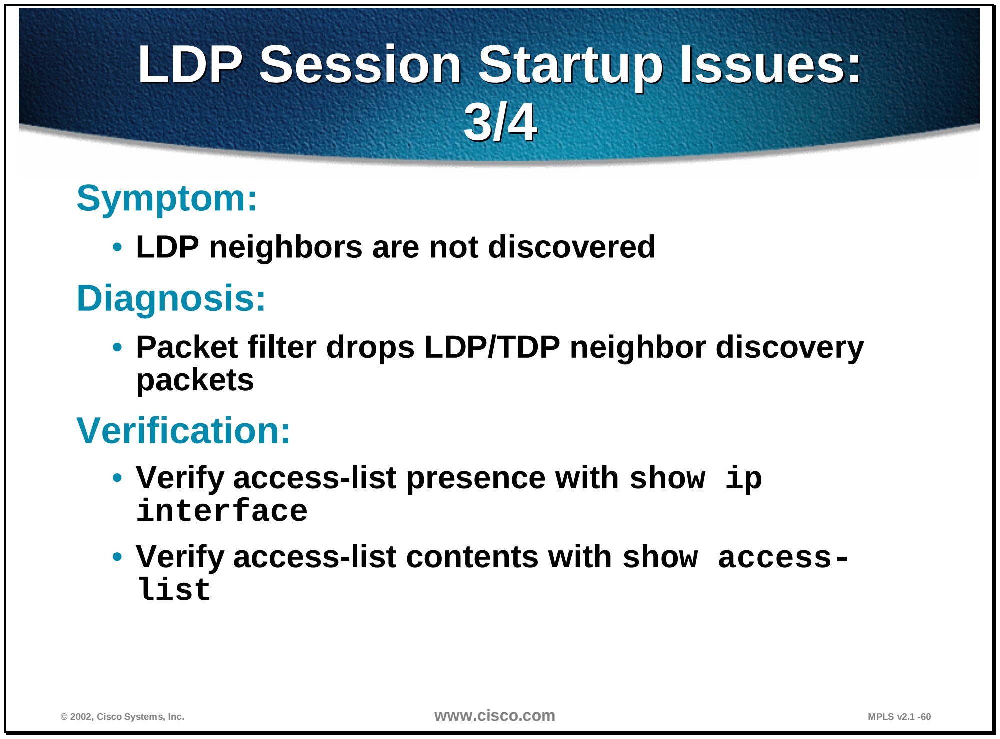
Solution: Use one of the following solutions:

a) Change the label distribution protocol on this end

b) Change the label distribution protocol on the other end

c) Enable both label distribution protocols on this end

d) Enable both label distribution protocols on the other end



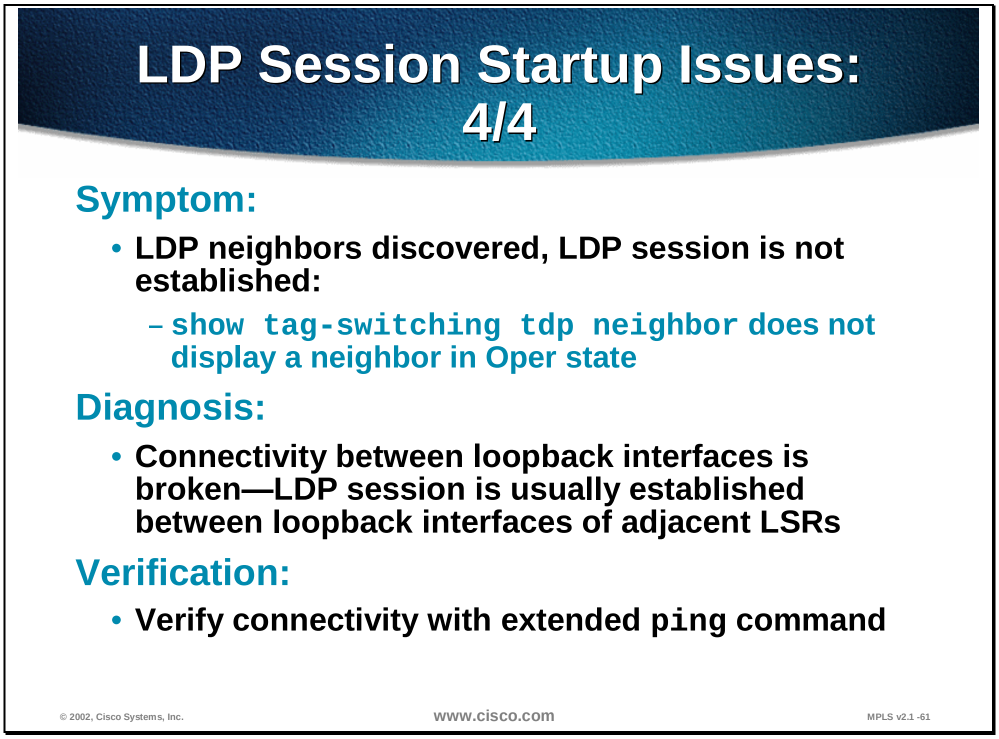
LDP (or TDP) Session Startup Problem

Symptom: LDP neighbors are not discovered.

Diagnose: MPLS configurations match on both ends but the session is still not established. Check whether there are any input access lists that deny discovery messages.

Verification: Use the **show ip interface** command to verify if there is any access-list in use on the interface. Then use the **show access-list** command to see which packets are permitted and denied.

Solution: Remove or change the access list to allow UDP packets with source and destination port number 646 for LDP and/or 711 for TDP. Make sure that the access list also allows TCP connections to and from port number 646 for LDP and/or 711 for TDP.



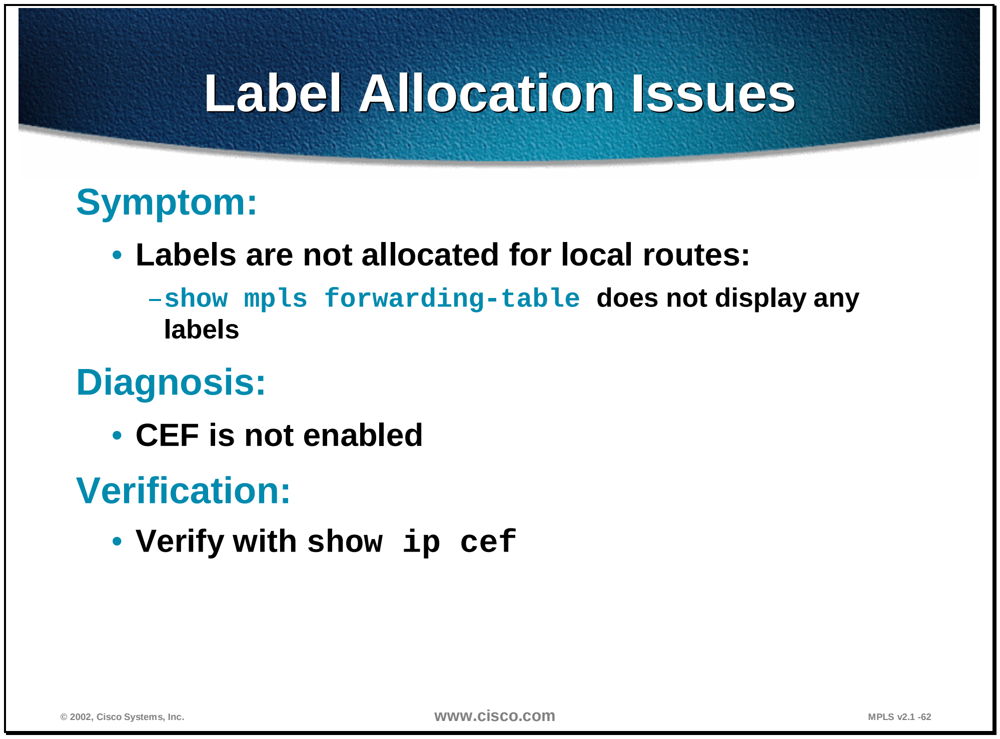
LDP (or TDP) Session Startup Problem

Symptom: LDP neighbors are not discovered. The **show tag-switching tdp neighbor** command does not display a neighbor in the **Oper** state.

Diagnose: LDP neighbors are exchanging hello packets but the LDP neighborship is never established. Connectivity between the loopback interfaces of the two routers is broken. The LDP session is usually established between loopback interfaces of adjacent LSRs.

Verification: Use the extended version of the **ping** command to check connectivity. The destination address of the **ping** command should of course be the neigbor’s loopback interface. But, more important, make user that the source address of the **ping** packets is the local router’s loopback interface.

Solution: Check the reachability of loopbacks as they are usually used to establish LDP neighborship. Make sure the loopback addresses are exchanged via the IGP used in the network.



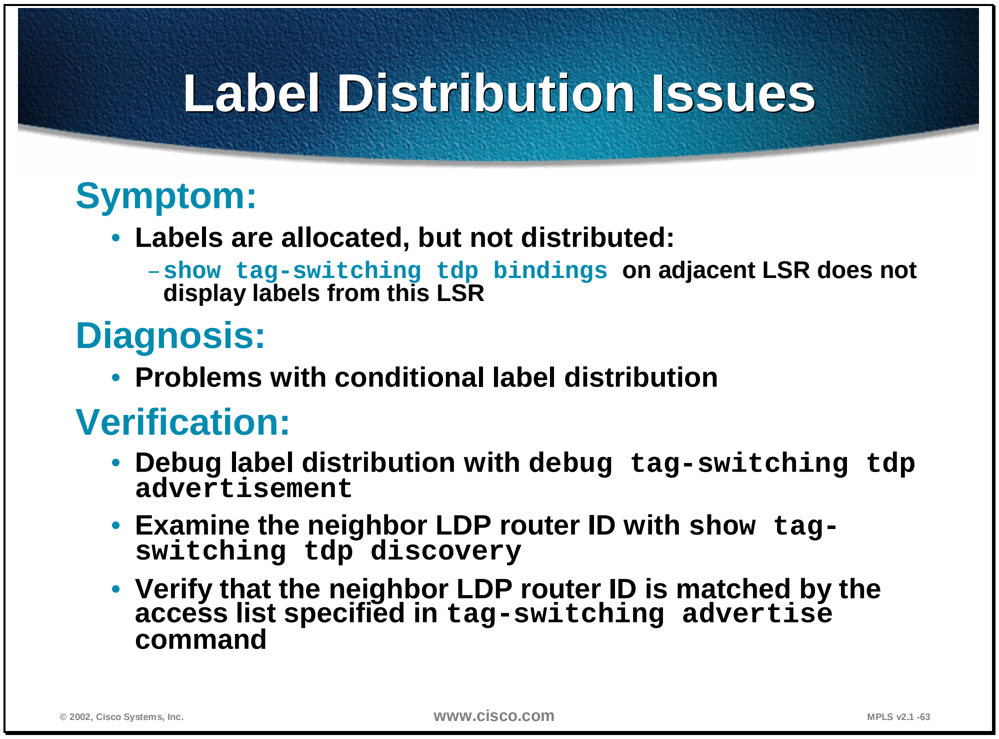
Labels Are Not Allocated

Symptom: Labels are not allocated for local routes. The **show mpls forwarding-table** does not display any labels.

Diagnose: Labels are not allocated for any or some of the local routes.

Verification: Use the **show ip cef** command to verify if CEF switching is enabled on all MPLS-enabled interfaces.

Solution: Enable CEF switching by using the **ip cef** global command or the **ip route-cache cef** interface command.



Labels Are Not Distributed

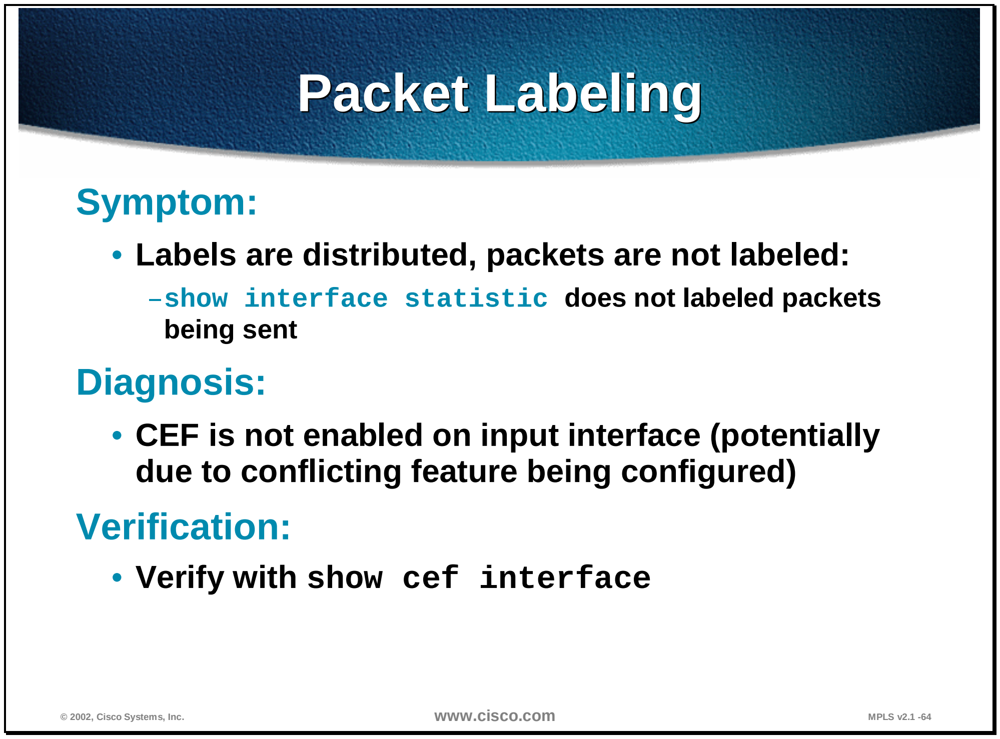
Symptom: Local labels are allocated but they are not propagated to the neighbor. The **show tag-switching tdp bindings** command on the neighboring router does not display any labels from the local router.

Diagnose: There is a problem with conditional label distribution.

Verification:

- Debug label distribution with **debug tag-switching tdp advertisement**
- Examine the neighbor LDP router ID with **show tag-switching tdp discovery**. Then verify that the neighbor LDP router ID is permitted by the access-list specifying neighbors in the **tag-switching advertise** configuration command.

Solution: Check whether conditional label advertising is enabled and verify both access lists used with the command **show access-list**.



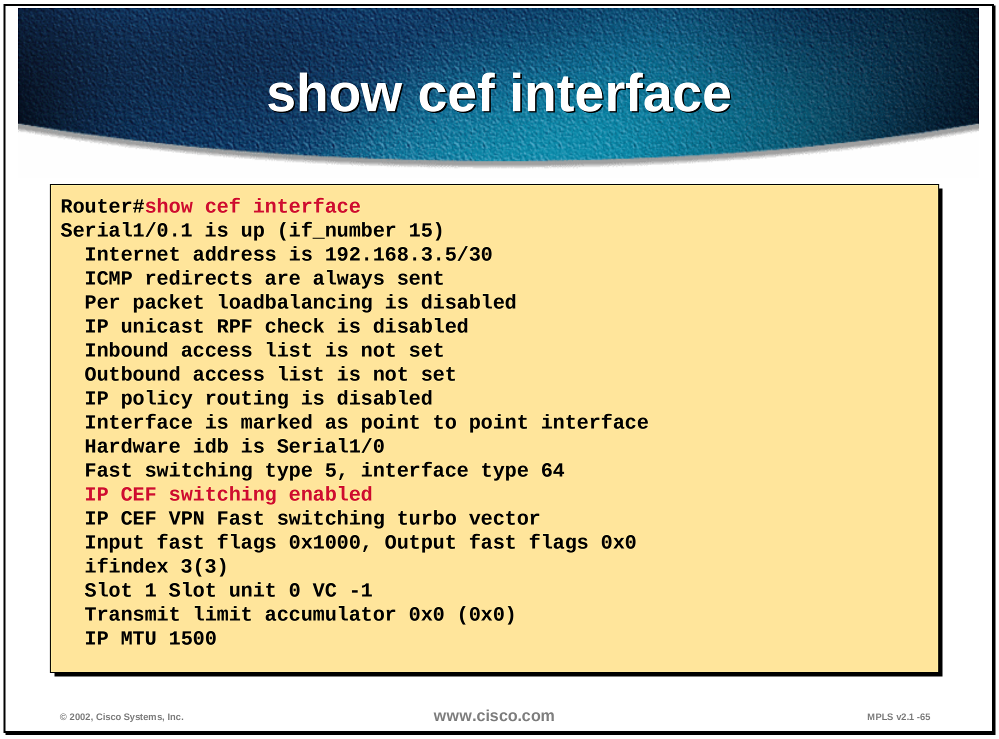
Forwarded Packets Are Not Labeled

Symptom: Labels are being distributed, but the packets forwarded are not labeled. The **show interface statistics** command dows not show any labeled packets being sent.

Diagnose: CEF is not enabled on the input interface. This could potentially be caused by conflicting features being configured on the interface.

Verification: Use the command **show cef interface** to verify whether CEF is used on the interface.

Solution: Enable CEF switching by using the **ip route-cache cef** interface command and make sure there is no feature enabled on the interface that is not supported in combination with CEF switching. Verify whether CEF is enabled on an individual interface with the **show cef interface** command.



show cef interface

To display CEF related interface information, use the **show cef interface** command in EXEC mode.

show cef interface *type number* [**detail**]

Syntax Description

type number	Interface type and number about which to display CEF related information.
detail	(Optional) Displays detailed CEF information for the specified interface type and number.

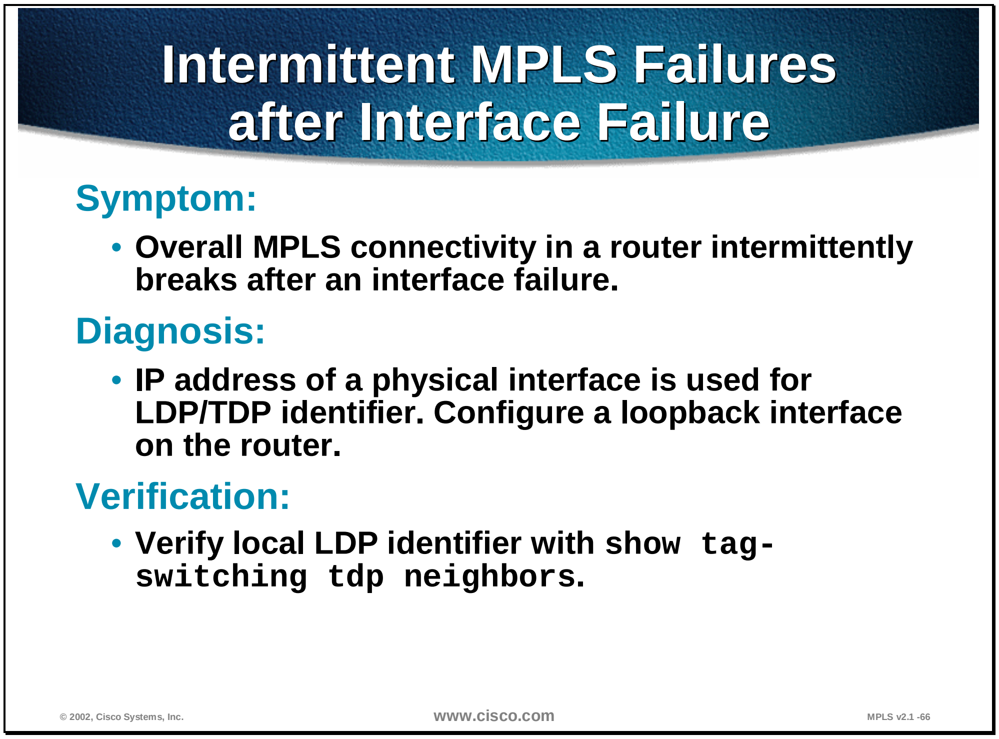
Usage Guidelines

This command is available on routers that have RP cards and line cards. The **detail** keyword displays more CEF-related information for the specified interface. You can use this command to show the CEF state on an individual interface.

The fields shown in the output are:

Field	**Description**
interface type number is {up | down}	Indicates status of the interface.
Internet address	Internet address of the interface.
ICMP packets are {always sent | never sent}	Indicates how packet forwarding is configured.

Per-packet load balancing	Status of load balancing in use on the interface (enabled or disabled).
Inbound access list {# | Not set}	Number of access lists defined for the interface.
Outbound access list	Number of access lists defined for the interface.
Hardware idb is *type number*	Interface type and number configured.
Fast switching type	Used for troubleshooting; indicates switching mode in use.
IP Distributed CEF switching {enabled | disabled}	Indicates the switching path used.
Slot *n* Slot unit *n*	The slot number.
Hardware transmit queue	Indicates the number of packets in the transmit queue.
Transmit limit accumulator	Indicates the maximum number of packets allowed in the transmit queue.
IP MTU	The value of the MTU size set on the interface.



MPLS Stops Working Intermittently After an Interface Failure

Symptom: MPLS connectivity is established, labels are exchanged and packets are labeled and forwarded as appropriate. However, an interface failure can sporadically stop an MPLS operation on unrelated interfaces in the same router.

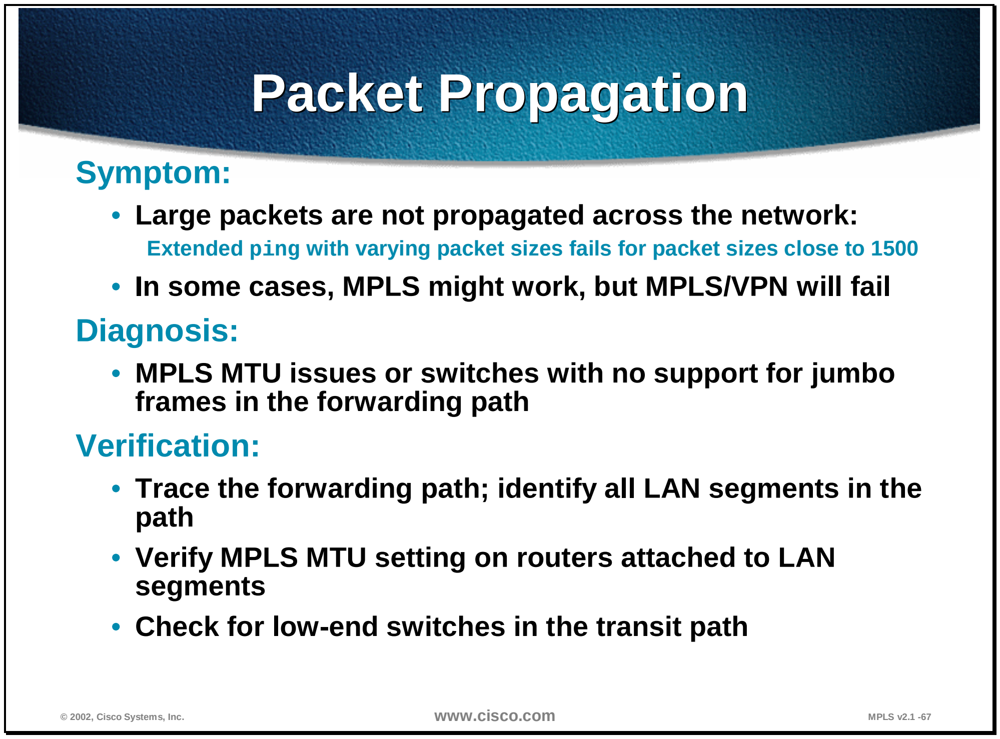
Details: TDP or LDP sessions are established between IP addresses that correspond to the TDP/LDP LSR identifier. The TDP/LDP LSR identifier is assigned using the algorithm that is also used to assign an OSPF or a BGP router identifier. This algorithm selects the highest IP address of an active interface if there are no loopback interfaces configured on the router. If that interface fails, the TDP/LDP LSR identifier is lost and the TCP session carrying TDP or LDP data is torn down, resulting in loss of all neighbor-assigned label information.

Diagnose: There is no loopback interface configured in any (ore both) of the two routers.

Verification: The symptom can be easily verified with **show tag-switching tdp neighbors** command, which displays the local and remote LSR identifiers. Verify that both of these IP addresses are associated with a loopback interface.

Solution: Configure a loopback interface on the LSR.

Note The TDP/LDP LSR identifier will only change after the router is reloaded.



Large IP Packets Are Not Propagated

Symptom: Large packets are not propagated across the network. The problem is more pronounced when more labels are used in a label stack as when MPLS/VPN and/or MPLS/TE is in use. Packets are labeled and sent but they are not received on the neighboring router.

Diagnose: MPLS MTU issues or/and a LAN switch between the adjacent MPLS-enabled routers may drop the packets if it does not support jumbo frames.

Verification:

Step 1 Trace the forwarding path. Identify all LAN segments in the path

Step 2 Verify MPLS MTU settings on routers attached to LAN segments

Step 3 Check for low-end switches in the transit path

Solution: Change the Tag Switching MTU, taking into account the maximum number of labels that may appear in a packet. Make sure that all LAN switches in the transit path support jumbo frames.

Summary

Show and debug commands are used to determine problems in a network. Use the **show mpls, show tag-switching** and **show ip cef** commands to diagnose problems in an MPLS-enabled network.

This module can be used as a troubleshooting guide to systematically pin down a problem in the shortest possible time.

---

*Use the debugging commands with extreme caution (particularly the **debug mpls packet** and the equivalent **debug tag-switching packet** command) because some debug commands generate a large amount of output and can cause a router to temporarily stop operating or even crash in a busy network.*

Lesson Review

1. Which command would you use to list LDP/TDP neighbors?
2. Which command will show you the LDP router ID of your neighbor?
3. Which command lists the contents of Label Information Base (LIB or TIB)?
4. Which command lists the contents of LFIB?
5. How would you display the label information attached to the packet during Layer 3 lookup?
6. Which debugging commands can be used to troubleshoot TDP session establishment?
7. Which commands can be used to troubleshoot label distribution?
8. What are the caveats against using the **debug mpls packet** command?

Summary

After completing this module, the learner will be able to perform the following tasks:

- Explain the basics of CEF switching
- Understand how to configure MPLS on frame-mode MPLS interfaces on IOS platforms
- Explain advanced core MPLS configuration options (TTL propagation, controlled label distribution) on IOS platforms
- Understand how to configure frame-mode MPLS on ATM PVC on IOS platforms
- Monitor and troubleshoot MPLS on IOS platforms